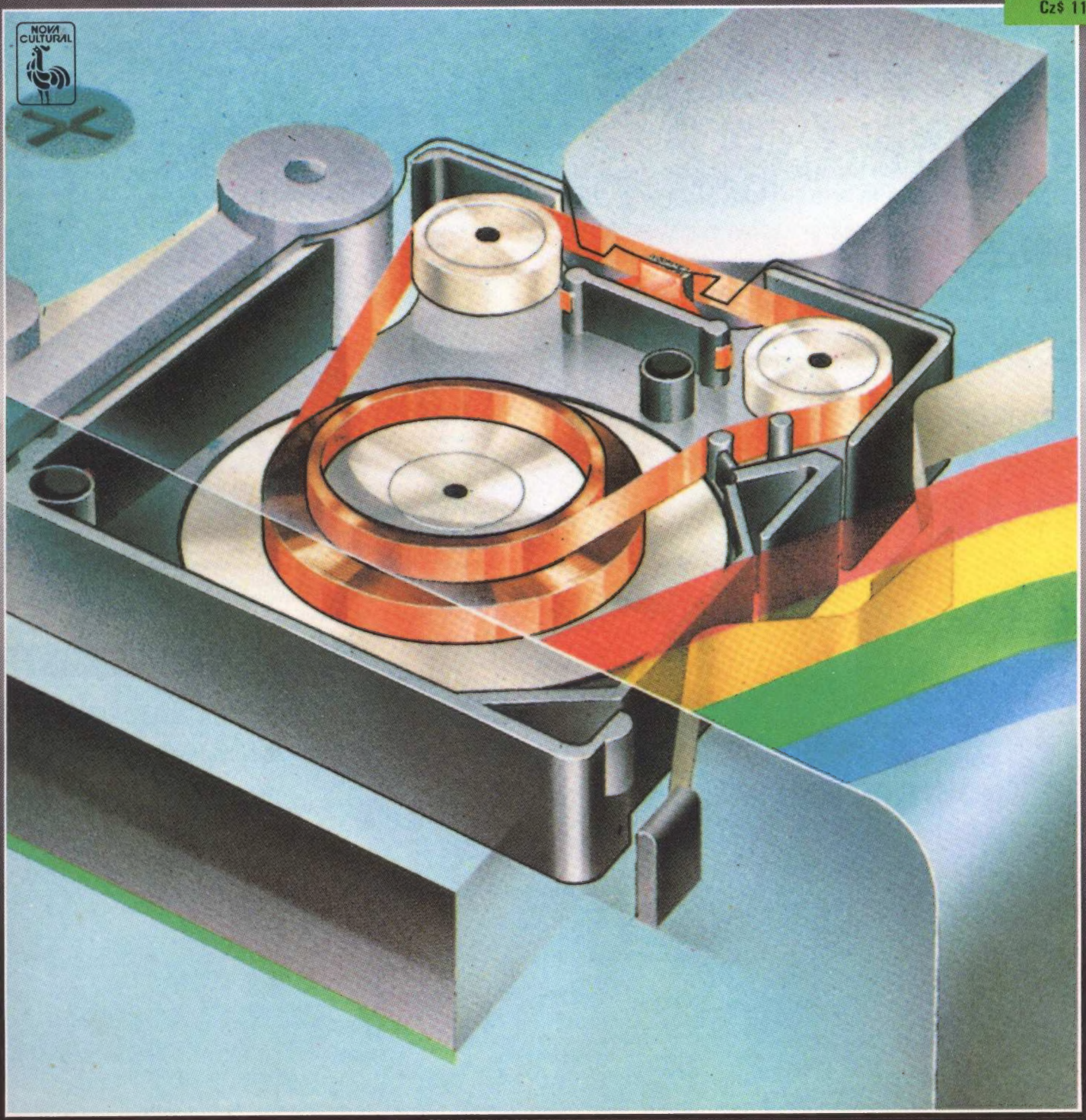
CURSO PRÁTICO 44 DE PROGRAMAÇÃO DE COMPUTADORES

INPUT

PROGRAMAÇÃO BASIC - PROGRAMAÇÃO DE JOGOS - CÓDIGO DE MÁQUINA

Cz$ 110,00



INPUT

Vol. 3 N.º 44

NESTE NÚMERO

PROGRAMAÇÃO BASIC

PROGRAMAÇÃO GRÁFICA DE CURVAS

Identificação das curvas. Elipse, parábola, hipérbole. Círculos e polígonos 861

PROGRAMAÇÃO BASIC

OS SEGREDOS DO SPECTRUM (1)

As linhas reservadas da tela. Delimitação das áreas independentes. Uma tela listrada 867

APLICAÇÕES

UMA AGENDA ELETRÔNICA (3)

Gravação das listas. Atualização. Impressão. Mudanças para sistemas com disquete 868

PROGRAMAÇÃO DE JOGOS

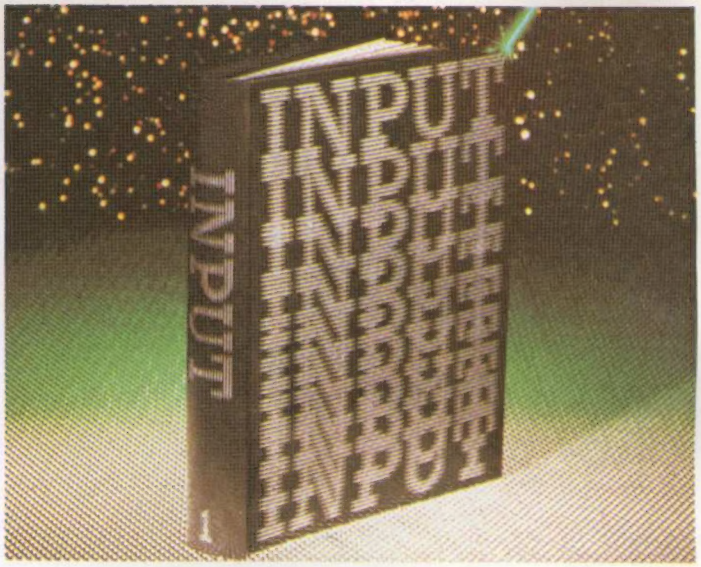
O JOGO A RAPOSA E OS GANSOS (1)

Teoria de programação. Resolução de problemas. Como avaliar posições no jogo 872

PERIFÉRICOS

A ESCOLHA DA MEMÓRIA AUXILIAR

Armazenamento em fita. Alças de fita. Acionadores de disco. Interfaces. Software 876



PLANO DA OBRA
INPUT é uma obra editada em fascículos semanais, e cada conjunto de 15 fascículos compõe um volume. A capa para encadernação de cada volume estará à venda oportunamente.

FÉRIAS, VIAGENS, MUDANÇAS...
NÃO FIQUE COM A COLEÇÃO INCOMPLETA
Se você está saindo de férias, pretende viajar ou vai se ausentar por algum tempo, avise antecipadamente seu jornaleiro. Ele pode guardar os seus fascículos enquanto você estiver fora. Se, por qualquer motivo, você perdeu alguns números, peça-os também a seu jornaleiro, ou entre em contato com nossa Distribuidora:

1. **Pessoalmente** — Em *São Paulo*, os endereços são: Rua Brigadeiro Tobias, 773, Centro, telefone 227-4188; Av. Industrial, 117, Santo André, telefone 449-0411, das 7h30 às 17h00 - dias úteis. No *Rio de Janeiro*, Av. Mem de Sá, 191/193, Centro, telefone (021) 222-7422, das 7h30 às 17h00 - dias úteis.
2. **Por carta** — Envie para:
   DINAP — Distribuidora Nacional de Publicações
   Números Atrasados
   Estrada Velha de Osasco, 132 — Jardim Teresa
   CEP 06040 — Osasco — SP
3. **Por telex** — Utilize o nº (11) 33 670 DNAP.

Em **Portugal**, os pedidos devem ser feitos à Distribuidora Jardim de Publicações Lda. — Qta. Pau Varais, Azinhaga de Fetais, 2685, Camarate, Lisboa; Apartado 57; Telex 43 069 JARLIS P.
Atenção: Após seis meses do encerramento da coleção, o atendimento dos pedidos dependerá da disponibilidade do estoque.
Obs.: Quando pedir livros, mencione sempre o título e/ou o autor da obra, além do número da edição.

COLABORE CONOSCO
Encaminhe seus comentários, críticas, sugestões ou reclamações ao
SERVIÇO DE ATENDIMENTO AO LEITOR
Caixa Postal 9 442, São Paulo — SP.

EDITOR
RICHARD CIVITA
NOVA CULTURAL

Presidente
Flávio Barros Pinto
Diretoria
Carmo Chagas, Iara Rodrigues
Pierluigi Bracco, Plácido Nicoletto,
Roberto Silveira, Shozi Ikeda,
Sônia Carvalho

REDAÇÃO
Diretor Editorial: Carmo Chagas
Editores Executivos:
Stefania Crema, Berta Sztark Amar
Editor Chefe: Paulo de Almeida
Editoras Assistentes: Ana Lúcia B. de Lucena,
Marisa Soares de Andrade
Chefe de Arte: Carlos Luiz Batista
Assistentes de Arte: Dagmar Bastos Sampaio,
Grace Alonso Arruda, Monica Lenardon Corradi

Secretário de Redação: Mauro de Queiroz

Colaboradores
Consultor Editorial Responsável:
Dr. Renato M. E. Sabbatini
(Diretor do Núcleo de Informática Biomédica da Universidade Estadual de Campinas)
Execução Editorial: DATAQUEST Assessoria em Informática Ltda., Campinas - SP
Tradução, adaptação, programação e redação:
Abílio Pedro Neto, Aluísio J. Dornellas de Barros,
Marcelo R. Pires Therezo, Marcos Huascar Velasco,
Raul Neder Porrelli, Ricardo J. B. de Aquino Pereira.
Coordenação Geral: Rejane Felizatti Sabbatini

COMERCIAL
Diretor Comercial: Roberto Silveira
Gerente Comercial: Joaquim Celestino da Silva
Gerente de Circulação: Denise Mozol
Gerente de Propaganda e Publicidade: José Carlos Madio
Gerente de Pesquisa e Análise de Mercado:
Wagner M. P. Nabuco de Araújo

CLC

A Editora Nova Cultural Ltda. é uma empresa do Grupo CLC - Comunicações, Lazer, Cultura S.A.
Presidente: Richard Civita
Diretoria: Flávio Barros Pinto, João Gomez, Menahem M. Politi, Renê C. X. Santos,
Stélio Alves Campos

© Marshall Cavendish Limited, 1984/85.
© Editora Nova Cultural Ltda., São Paulo, Brasil, 1986, 2ª edição, 1987.
Edição organizada pela Editora Nova Cultural Ltda.
Av. Brigadeiro Faria Lima, 2000 - 3º andar
CEP 01452 - São Paulo - SP - Brasil
(Artigo 15 da Lei 5 988, de 14/12/1973).
Esta obra foi composta pela AM Produções Gráficas Ltda. e impressa pela Companhia Lithographica Ypiranga.

PROGRAMAÇÃO GRÁFICA DE CURVAS

- IDENTIFICAÇÃO DAS CURVAS
- UMA ELIPSE NO COTIDIANO
- NADO PARABÓLICO
- CÍRCULOS E POLÍGONOS
- GRÁFICOS FEITOS DE CURVAS

No primeiro artigo sobre seções cônicas, mostramos como desenhar círculos, elipses, parábolas e hipérboles no micro. Agora, veremos como incorporá-los aos programas.

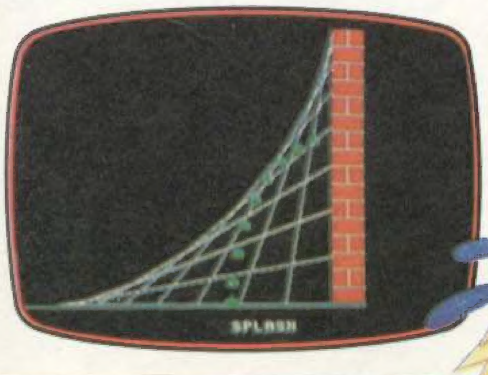
Como vimos em alguns exemplos do artigo anterior (página 801), as curvas cônicas estão no nosso dia-a-dia, por toda a parte, e muitas vezes nem chegamos a percebê-las. Mas não é difícil identificá-las: basta recorrer às equações. Se, ao calcularmos a posição de um objeto em movimento, verificarmos que sua coordenada **X** é dada por **A*COS T** e sua coordenada **Y**, por **A*SIN T** (onde **A** é uma distância fixa e **T**, um ângulo que varia), ficará absolutamente evidente que estamos lidando com a equação de um círculo.

Por outro lado, algumas vezes é mais fácil analisar o modo como algo se move do que calcular suas equações. Se, por exemplo, observarmos que a distância de um objeto a um ponto é sempre constante, saberemos que se trata de um círculo, sem precisarmos calcular sua equação. Esse tipo de análise também se aplica a outras curvas.

ELIPSE, PARÁBOLA, HIPÉRBOLE

Uma elipse não difere muito de um círculo e, assim como este, é de fácil identificação. Ela é desenhada quando

À medida que a escada escorrega, o balde preso a ela traça parte de uma elipse.



um ponto se move de tal maneira que a sua distância a um dos focos somada à sua distância ao outro foco permanece sempre constante.

Uma parábola, por sua vez, é desenhada quando um ponto se move de tal modo que a sua distância a um ponto fixo é igual à distância perpendicular a uma linha fixa. O ponto fixo corresponde ao foco da parábola. A linha fixa, chamada de *diretriz*, é uma reta perpendicular ao eixo de simetria da parábola e que não corta a mesma — ou seja, está fora dela. Como já vimos, a distância da diretriz a um ponto qualquer da curva é igual à distância desse mesmo ponto até o foco.

A descrição da hipérbole também é simples: ela é obtida quando um ponto se move e a distância dele a um ponto fixo menos sua distância a outro ponto fixo permanece constante. Os pontos fixos, um em cada metade da hipérbole, são seus focos.

Os programas apresentados a seguir demonstram os dois métodos de identificação de curvas — isto é, o de equações e o de observação do comportamento de um ponto em movimento. Uma vez percebida a presença de um certo tipo de curva num programa, fica bem mais fácil utilizá-lo.

UMA ESCADA ESCORREGANDO

O primeiro programa mostra parte de uma elipse obtida durante um evento comum: uma escada escorregando numa parede. Observamos que um balde preso à escada traça parte de uma elipse à medida que esta escorrega.

Suponhamos que a escada tem um comprimento de oitocentas unidades e que o balde está preso a quinhentas unidades do pé da escada. A posição do balde será **X = -300*COS (ângulo)** e **Y = 500*SIN (ângulo)**, o que sabemos corresponder à equação de uma elipse.

```
10  HOME :C =  ATN (1) / 45
15  HGR : HCOLOR= 3
20  GOSUB 250
30  GOSUB 50
40  GOTO 40
50  REM  ESCADA
60  FOR A = 80 TO 0 STEP  - 10
70  FOR T = 0 TO 250: NEXT
80  HPLOT 230 - 150 *  COS (C *
 A),150 TO 228,150 - 150 *  SIN
 (C * A)
110 X =  - 56 *  COS (C * A)
120 Y = 90 *  SIN (C * A)
130  GOSUB 190
140  PRINT  CHR$ (7)
150  NEXT
190  REM  BALDE
200  IF Y = 0 THEN  VTAB (22):
HTAB (22): PRINT "SPLASH"
210  HPLOT 228 + X,154 - Y TO 2
28 + X,150 - Y
220  HPLOT  TO 232 + X,150 - Y
230  HPLOT  TO 232 + X,154 - Y
240  HPLOT  TO 228 + X,154 - Y:
 RETURN
250  REM  PAREDE
260 OX = 230
270  HPLOT OX,0 TO OX,150
280  HPLOT OX + 8,0 TO OX + 8,1
50
290  HPLOT OX + 16,0 TO OX + 16
,150
300  FOR Y = 0 TO 150 STEP 10:
HPLOT OX,Y TO OX + 16,Y
310  NEXT
320  RETURN
```

```
 10 LET wall=240: LET ladder=
 60: LET bucket=190
 20 GOSUB wall
 30 GOSUB ladder
 35 FLASH 0
 40 GOTO 40
 60 FOR a=80 TO 0 STEP -10
 70 PAUSE 25: LET r=a/(180/PI)
 80 PLOT ox-150*COS (r),oy
 90 DRAW ox-(ox-150*COS (r)),
 oy+150*SIN (r)
110 LET x=-60*COS (r)
120 LET y=90*SIN (r)
130 GOSUB bucket
140 SOUND .1,a/2-15
150 NEXT a
160 FLASH 1: PRINT AT 10,5;"SP
LASH"
170 RETURN
190 PLOT ox+x,oy+y+5: DRAW 0,-
2
200 FOR n=oy+y TO oy+y+2: PLOT
ox+x-2,n: DRAW 4,0
210 NEXT n
220 RETURN
240 BORDER 0: INK 7: PAPER 0:
CLS
250 LET ox=232: LET oy=8
260 FOR y=1 TO 20: PRINT
PAPER 2;AT y,29;"  "
270 NEXT y
280 FOR y=oy-1 TO 165 STEP 16:
PLOT ox,y
290 DRAW 16,0: PLOT ox,y+8:
DRAW 16,0: PLOT ox+8,y+8: DRAW
0,8
300 NEXT y
310 PLOT  INK 4;ox+8,oy-1:
DRAW  INK 4;-232,0
320 RETURN
```

```
10 COLOR 15,4,5:SCREEN2
15 C=ATN(1)/45
20 GOSUB 230:REM<parede>
30 GOSUB 50:REM<escada>
40 GOTO 40
50 FOR AN=80 TO 0 STEP-10
90 LINE(230-150*COS(C*AN),150)-
(228,150-150*SIN(C*AN)),15
110 X=-56*COS(C*AN)
120 Y=90*SIN(C*AN)
130 GOSUB 200:REM<balde>
140 FORT=0 TO 500:NEXT
150 NEXT
160 RETURN
200 IF Y=0 THEN Y=4:DRAW"BM160,
156C1S16LDRDLBR2U2RDLBEBRD2RBRU
2RDNLDBRRULURBRD2BRUNLUC4"
210 LINE(228+X,154-Y)-(232+X,15
0-Y),15,BF
220 RETURN
230 LINE(230,0)-(255,150),6,BF
250 FOR Y=0 TO 150 STEP 10
260 LINE(230,Y)-(255,Y),10
270 NEXT
```

```
280 FOR Y=0 TO 150 STEP 20
290 LINE(243,Y)-(243,Y+10),10
300 NEXT
310 LINE(0,151)-(255,191),3,BF
320 RETURN
```

```
10 PMODE 3,1:PCLS:SCREEN 1,0:C=
ATN(1)/45
20 GOSUB 230
30 GOSUB 50
40 GOTO 40
50 FOR AN=80 TO 0 STEP -10
70 COLOR 4,2
90 LINE(230-150*COS(C*AN),150)-
(228,150-150*SIN(C*AN)),PRESET
110 X=-56*COS(C*AN)
120 Y=90*SIN(C*AN)
130 GOSUB 200
140 FOR T=0 TO 500:NEXT
150 NEXT
160 RETURN
200 IF Y=0 THEN Y=4:DRAW"BM160,
156C2S16LDRDLBR2U2RDLBEBRD2RBRU
2RDNLDBRRULURBRD2BRUNLUC4"
210 LINE(228+X,154-Y)-(232+X,15
0-Y),PSET,BF
220 RETURN
230 LINE(230,0)-(255,150),PSET,
BF
240 COLOR 2
250 FOR Y=0 TO 150 STEP 10
260 LINE(230,Y)-(255,Y),PSET
270 NEXT
280 FOR Y=0 TO 150 STEP 20
290 LINE(243,Y)-(243,Y+10),PSET
300 NEXT
310 COLOR 3:LINE(0,151)-(255,19
1),PSET,BF
320 RETURN
```

O programa compõe-se de três rotinas principais — a que desenha a parede, a que desenha a escada e a que desenha o balde. A parede é traçada primeiro, entre as linhas 230 (250 no Apple) e 320. Depois, a rotina entre as linhas 50 e 160 desenha a escada em nove posições diferentes a intervalos de 10 graus. Essa rotina chama a seguinte, que fica entre as linhas 190 e 220 (240 no Apple) e tem a função de desenhar um balde para cada posição da escada. As coordenadas do balde são calculadas nas linhas 110 e 120, e, como vimos, elas formam a equação de uma elipse. As posições anteriores do balde e da escada não são apagadas, para que se perceba mais facilmente que o caminho percorrido pelo balde realmente faz parte de uma elipse.

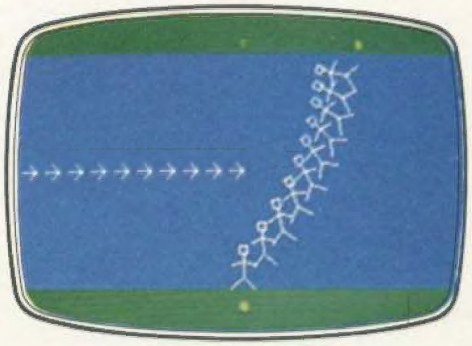
NADO PARABÓLICO

Imagine o que acontece quando um nadador tenta atravessar um rio de águas rápidas. Mesmo que ele tenha como alvo um determinado ponto na margem oposta, a correnteza certamente o desviará um pouco do seu destino. Caso a velocidade da correnteza se equipare à do nadador, a distância do desvio será igual à metade da largura do rio, e o efeito conjunto das duas velocidades fará com que o nadador percorra um caminho parabólico.

Para entendermos a razão por que isso ocorre, devemos pensar em termos de velocidade. O nadador visa sempre um ponto na outra margem e nada com uma velocidade **V**, enquanto o rio corre paralelamente à margem, com a mesma velocidade **V**. Combinando os dois valores, obtemos a velocidade real do nadador em relação à margem.

Esse raciocínio baseia-se na regra do paralelogramo de forças, empregada em problemas de física. Ela é usada na construção de uma parábola, onde a distância do nadador ao foco (ponto que ele visa na outra margem) é igual à dis-

Um nadador que se move com a velocidade da correnteza do rio percorre uma trajetória parabólica.



tância dele até a diretriz (distância percorrida pelo rio no mesmo intervalo de tempo).

```
10  HGR2 : HCOLOR= 3
20  GOSUB 50
30  GOSUB 200
40  GOTO 40
50  REM  DESENHA RIO
70  FOR LY = 0 TO 38: HPLOT 0,L
Y TO 279,LY: NEXT
80  FOR LY = 153 TO 191: HPLOT
0,LY TO 279,LY: NEXT
90  RETURN
200  FOR T =  - 1 TO  - .05 STE
P .1
210 X =  - 60 * T * T:Y = 120 *
 T
220 AN =  ATN ((X + 60) /  - Y)

230  GOSUB 300
240 FX =  INT (130 + Y): HPLOT
FX,95 TO FX + 8,95
245  HPLOT FX + 4,91 TO FX + 8,
95
250  HPLOT  TO FX + 4,99
255  NEXT
260  RETURN
300 XC = 187 + X:YC = 33 - Y
310 XX = 0:YY = 6: GOSUB 420
320  HPLOT  INT (XC + XT), INT
(YC + YT)
330  RESTORE
340  FOR N = 1 TO 16
350  READ XX,YY
360  GOSUB 420
370  HPLOT  TO XC + XT + .5,YC
- YT + .5
380  NEXT
390  RETURN
410  DATA  0,6,-6,0,0,6,6,0,0,6
,0,14,6,12,0,14,-6,12,0,14,0,18
```

CONDIÇÕES DE VISUALIZAÇÃO

Várias providências podem ser tomadas para aperfeiçoar as condições de visualização do vídeo. Por exemplo, o esforço visual será diminuído com o uso de caracteres claros sobre fundo escuro. Assim, procure "sintonizar" pouco a pouco os controles de contraste e intensidade, até melhorar a definição dos caracteres.

Experimente posicionar sua mesa de trabalho de tal maneira que a tela não reflita a luz proveniente de janelas ou de luminárias. Incline ligeiramente a tela, dirigindo os reflexos para longe de seus olhos. Também é aconselhável cobrir a tela com um filtro anti-reflexivo (à venda em lojas de computadores).

```
,2,18,2,22,-2,22,-2,18,0,18
420 XT = XX *  COS (AN) - YY *
 SIN (AN)
430 YT = XX *  SIN (AN) + YY *
 COS (AN)
440  RETURN
```

```
 10 BORDER 0: PAPER 0: INK 7:
 CLS
 20 REM rio
 60 LET parabola=190: LET swim
 mer=300: LET rotate=430
 70 LET a$="
                "
 80 FOR n=0 TO 3
 90 PRINT  PAPER 4;a$: NEXT n
100 FOR n=4 TO 18
110 PRINT  PAPER 1;a$: NEXT n
140 PRINT  PAPER 2; INK 6;AT 3
,15;"F";AT 19,15;"A";AT 3,22;"
O"
150 GOSUB parabola
160 STOP
190 LET ox=187: LET oy=150
200 FOR t=-1 TO -0.05 STEP 0.1
210 LET x=-60*(t*t): LET y=120
*t
220 LET a=ATN ((x+60)/-y)
230 PLOT ox-60+y,oy-60: DRAW
INK 7;10,0: DRAW  INK 7;-5,-5:
DRAW  INK 7;0,10: DRAW  INK 7;
5,-5
240 GOSUB swimmer
250 NEXT t
260 RETURN
300 LET ox=ox+x: LET oy=oy+y
310 LET x=0: LET y=6
320 GOSUB rotate
330 PLOT ox+xt,oy+yt
340 RESTORE 410
350 FOR n=1 TO 17
360 READ x,y
370 GOSUB rotate
380 DRAW xt,yt
390 NEXT n
400 LET ox=188: LET oy=150:
RETURN
410 DATA -3,0,0,3,3,0,0,-3,-2,
0,0,-3,-4,0,0,4,0,-4,8,0,0,4,0
,-4,-4,0,0,-4,-4,-4,4,4,4,-4
430 LET xt=x*COS (a)-y*SIN (a)
440 LET yt=x*SIN (a)+y*COS (a)
450 RETURN
```

```
10 COLOR 15,9,9:SCREEN2
20 GOSUB 50
30 GOSUB 200
40 GOTO 40
50 LINE(0,38)-(255,153),5,BF
60 DRAW"BM123,158C15S16ND2RDNLD
"
70 DRAW"BM123,25NRDNRD"
80 DRAW"BM180,25RD2LU2"
90 RETURN
200 FOR T=-1 TO -.05 STEP .1
210 X=-60*T*T:Y=120*T
220 AN=ATN((X+60)/-Y)
230 GOSUB 300
240 DRAW"BM"+STR$(INT(130+Y))+"
,95"+"C9R2NGH"
250 NEXT
260 RETURN
300 XC=187+X:YC=33-Y
310 XX=0:YY=6:GOSUB 420
320 DRAW"BM"+STR$(INT(XC+XT))+"
,"+STR$(INT(YC-YT))
330 RESTORE
340 FOR N=1 TO 16
350 READ XX,YY
360 GOSUB 420
370 LINE-(XC+XT+.5,YC-YT+.5),15
380 NEXT
390 RETURN
410 DATA 0,6,-6,0,0,6,6,0,0,6,0
,14,6,12,0,14,-6,12,0,14,0,18,2
,18,2,22,-2,22,-2,18,0,18
420 XT=XX*COS(AN)-YY*SIN(AN)
430 YT=XX*SIN(AN)+YY*COS(AN)
440 RETURN
```

```
10 PMODE 3,1:PCLS:SCREEN 1,0
20 GOSUB 50
30 GOSUB 200
40 GOTO 40
50 COLOR 3,2:LINE(0,38)-(255,15
3),PSET,BF
60 DRAW"BM123,158C4S16ND2RDNLD"
70 DRAW"BM123,25NRDNRD"
80 DRAW"BM180,25RD2LU2"
90 RETURN
200 FOR T=-1 TO .05 STEP .1
210 X=-60*T*T:Y=120*T
220 AN=ATN((X+60)/-Y)
230 GOSUB 300
240 DRAW"BM"+STR$(INT(130+Y))+"
,95"+"C2R2NGH"
250 NEXT
260 RETURN
300 XC=187+X:YC=33-Y
310 XX=0:YY=6:GOSUB 420
320 DRAW"BM"+STR$(INT(XC+XT))+"
,"+STR$(INT(YC-YT))
330 RESTORE
340 FOR N=1 TO 16
350 READ XX,YY
360 GOSUB 420
370 LINE -(XC+XT+.5,YC-YT+.5),P
RESET
380 NEXT
390 RETURN
410 DATA 0,6,-6,0,0,6,6,0,0,6,0
,14,6,12,0,14,-6,12,0,14,0,18,2
,18,2,22,-2,22,-2,18,0,18
420 XT=XX*COS(AN)-YY*SIN(AN)
430 YT=XX*SIN(AN)+YY*COS(AN)
440 RETURN
```

A primeira seção do programa, nas linhas 50 a 160, desenha o rio e as margens. A seção seguinte, nas linhas 170 a 290, usa a equação da parábola (linha 210) para calcular a posição do nadador. A rotina do nadador está entre as linhas 300 e 400. Ela recorre à rotina de rotação (linhas 420 a 450) para garantir que o nadador seja desenhado com o ângu-

lo correto, ou seja, para que esteja visando sempre o mesmo ponto na outra margem (que, neste caso, corresponde ao foco da parábola). O formato do nadador é definido pela linha **DATA** 410.

CÍRCULOS E POLÍGONOS

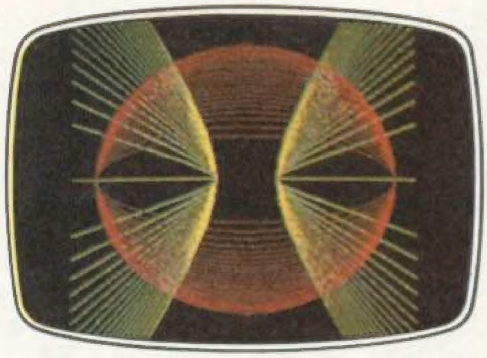
Para o computador, um círculo nada mais é do que um polígono de muitos lados. Quanto maior for o número de lados, mais lisa será a curva. Os usuários do Apple já devem ter percebido que temos usado polígonos para obter círculos, compensando a lamentável falta do comando **CIRCLE**. No programa a seguir, eles verão que as linhas 70, 75 e 80 simulam um comando **CIRCLE** para o desenho de um círculo de raio 70 e com centro em 127,95. Por uma questão visual, fizemos o raio na direção **Y** um pouco menor (65 em vez de 70). Caso contrário, o círculo ficaria oval, aparentando ser uma elipse.

```
10 PI = 4 *  ATN (1)
60  HOME : HGR : HCOLOR= 1
69  HPLOT 197,85
70  FOR TH = 0 TO 2 * PI STEP P
I / 16
75 CX = 70 *  COS (TH) + 127:CY
 = 65 *  SIN (TH) + 95
80  HPLOT  TO CX,CY: NEXT
90  VTAB (23): INPUT "FORNECER
ANGULO ";A
100 A = A *  ATN (1) / 45
110  GOSUB 260
130  HOME : VTAB (23): INPUT "J
OGA NOVAMENTE (S/N) ";AN$
140  IF AN$ = "S" THEN 10
150  IF AN$ <  > "N" THEN 130
160  HOME : TEXT : END
260 TH = 2 * A
270 N = 0
280  HPLOT 185,95
300  HPLOT  TO 127 + 58 *  COS
(TH),95 - 58 *  SIN (TH)
310 TH = TH + 2 * A
330 N = N + 1
340  IF N < 15 THEN 300
350  RETURN
```

```
10 LET ox=100: LET oy=90
20 LET polygon=260
30 PAPER 0: INK 6
40 BORDER 0
50 CLS
80 PRINT  INK 7;AT 0,22;"Raio
 do"'TAB 22;"circulo e"'TAB
22;"82 unid."
90 PRINT  INK 4;AT 21,0;"Intr
oduza o angulo";
 95 CIRCLE 100,90,82
100 INPUT a
105 PRINT AT 21,0;"
"
110 GOSUB polygon
120 PRINT AT 21,6;"Outra vez ?
(S/N)"
130 LET a$=INKEY$: IF a$="s"
THEN  GOTO 20
140 IF a$="n" THEN  STOP
150 GOTO 130
260 LET a=a/(180/PI)
270 LET at=0
280 LET t=2*a
290 PLOT ox+82,oy
300 FOR n=0 TO 15
310 LET b=82*COS (t)+ox: LET c
=82*SIN (t)+oy
320 LET b=(b-(PEEK 23677)):
LET c=(c-(PEEK 23678)): DRAW b
,c
330 SOUND 0.01,(n*5)-20
340 LET t=t+2*a
350 NEXT n
360 RETURN
```

```
10 COLOR1,15,15
20 GOSUB 90
30 GOSUB 60
40 GOSUB 120
50 CLS:END
60 SCREEN2
70 CIRCLE(127,95),70,13:CIRCLE(
127,95),60,13:PAINT(127,30),13
80 GOTO 260
90 CLS:LOCATE9,10:INPUT"Forneça
 um ângulo ";A
100 A=A*ATN(1)/45
110 RETURN
120 SCREEN0:CLS
130 LOCATE5,10:INPUT"Joga novam
ente (s/n) ";AN$
140 IF AN$="S" OR AN$="s" THEN
RUN
150 IF AN$<>"N" AND AN$<>"n" TH
EN 120
160 RETURN
260 TH=2*A
270 N=0
280 DRAW"BM185,95"
300 LINE-(127+58*COS(TH),95-58*
SIN(TH)),1
310 TH=TH+2*A
320 PLAY"T255L64V"+STR$(15-N)+"
CAGF"
330 N=N+1
340 IF N<15 THEN 300
350 IF INKEY$="" THEN 350
360 RETURN
```

```
10 PMODE 3,1
60 PCLS:SCREEN 1,0
70 CIRCLE(127,95),70,4:CIRCLE(1
27,95),60,4:PAINT(127,30),4
80 FOR G=1 TO 2000:NEXT:COLOR 2
90 CLS:PRINT:INPUT"QUAL O ANGUL
O ";A
```



Efeitos muito interessantes podem ser obtidos pela sobreposição de curvas cônicas.

```
100 A=A*ATN(1)/45
110 SCREEN 1,0:GOSUB 260
120 IF INKEY$="" THEN 120
130 PRINT:PRINT:INPUT"NOVAMENTE
 (S/N) ";AN$
140 IF AN$="S" THEN 60
150 IF AN$<>"N" THEN 130 ELSE C
LS:END
260 TH=2*A
270 N=0
280 DRAW"BM185,95"
300 LINE-(127+58*COS(TH),95-58*
SIN(TH)),PSET
310 TH=TH+2*A
320 PLAY"T20V"+STR$(31-N*2)+"C"
330 N=N+1
340 IF N<15 THEN 300 ELSE RETUR
N
```

O programa desenha um círculo e pede que forneçamos o ângulo que a primeira linha formará com o lado do círculo. O ângulo **A** ou **a** é convertido para *Teta* (**TH** ou **t**) na linha 260 da rotina que desenha o polígono. O número de lados foi restrito a quinze (linha 340) para o diagrama não ficar parecendo apenas um emaranhado de linhas.

Se fornecermos um ângulo pequeno, o polígono se assemelhará a um círculo. Com ângulos maiores, a forma mais provável será a de uma estrela.

ARTE POR COMPUTADOR

O programa a seguir desenha hipérboles com diferentes excentricidades e, sobre elas, várias elipses. Podemos combinar outros tipos de curva para obter desenhos mais complexos — basta dar asas à imaginação.

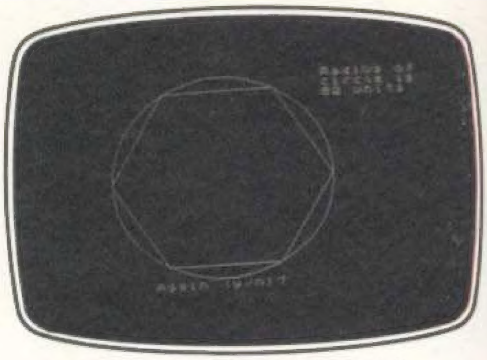
```
10  HOME : HGR2 : HCOLOR= 3
20 C =  ATN (1) / 45:PI = 4 *
ATN (1)
30  GOSUB 70
40  GOSUB 255
```

```
50  GOTO 50
70  FOR E = 1 TO 1.25 STEP .02
100 A = 22:B = A *  SQR (E * E
- 1)
110  HPLOT 128 +  INT (A /  COS
 ( - 80 * C)),95 -  INT (B *  T
AN ( - 80 * C))
130  FOR TH =  - 80 TO 80 STEP
20
140 X = A /  COS (TH * C)
150 Y = B *  TAN (TH * C)
160  HPLOT  TO 128 + X,95 - Y
170  NEXT
180  HPLOT 127 +  INT (A /  COS
 (100 * C)),95 -  INT (B *  TAN
 (100 * C))
190  FOR TH = 100 TO 260 STEP 2
0
200 X = A /  COS (TH * C)
210 Y = B *  TAN (TH * C)
220  HPLOT  TO 127 + X,95 - Y
230  NEXT TH,E
250  RETURN
255  HCOLOR= 2: HPLOT 222,95
260  FOR E = 1 TO .1 STEP  - .0
3
270  FOR AN = 0 TO 2 * PI STEP
PI / 16
280 CX = 95 *  COS (AN) + 127:C
Y = 90 * E *  SIN (AN) + 95
290  HPLOT  TO CX,CY: NEXT AN
300  NEXT E: RETURN
```

```
 10 BORDER 0: PAPER 0: INK 7:
 CLS
 20 LET hiperbola=80
 30 LET elipse=270
 40 GOSUB hiperbola
 50 GOSUB elipse
 60 GOTO 60
 80 LET ox=128: LET oy=87
 90 FOR e=1 TO 2 STEP 0.05
100 LET a=22: LET b=a*(SQR (e^
2-1))
102 LET h=1
104 LET f=ox+(a/COS (-1.396))
106 LET g=oy+(b*TAN (-1.396))
108 IF g<0 THEN  LET h=0
110 PLOT  INVERSE 1; OVER 1;f,
h
120 IF g>0 THEN  PLOT  INK 6;f
,g
130 FOR t=-80 TO 80 STEP 20
135 LET r=t/(180/PI)
140 LET x=a/COS (r): LET y=b*
TAN (r)
142 LET c=oy+y: LET d=ox+x
150 IF h=0 THEN  LET d=f+g*(f-
d)/(c-g): PLOT d,h: LET c=0
160 IF c>175 THEN  LET d=d-((d
-PEEK 23677)*(c-175)/(c-PEEK
23678)): LET c=175
170 DRAW  INK 6;d-PEEK 23677,c
-PEEK 23678: NEXT t
172 LET f=ox+(a/COS (1.75))
174 LET g=oy+(b*TAN (1.75))
176 PLOT  INVERSE 1; OVER 1;f,
h
178 IF g<0 THEN  LET h=0
180 IF g>0 THEN  PLOT  INK 6;f
,g
190 FOR t=100 TO 260 STEP 20
195 LET r=t/(180/PI)
200 LET x=a/(COS (r)): LET y=b
*TAN (r)
.202 LET c=oy+y: LET d=ox+x
204 IF h=0 THEN  LET d=f+g*(f-
d)/(c-g): PLOT d,h: LET c=0
206 LET h=1
210 IF c>175 THEN  LET d=0-((d
-PEEK 23677)*(c-175)/(c-PEEK
23678)): LET c=175
220 DRAW  INK 6;d-PEEK 23677,c
-PEEK 23678
230 NEXT t: NEXT e
250 RETURN
270 FOR e=0.5 TO 0.98 STEP
0.04
280 LET a=100: LET b=a*(SQR (1
-e^2))
290 PLOT ox+a,oy
300 FOR t=0 TO 360 STEP 10
305 LET r=t/(180/PI)
310 LET x=a*COS (r)
320 LET y=b*SIN (r)
330 DRAW x-(PEEK 23677)+ox,y-(
PEEK 23678)+oy
340 NEXT t: NEXT e
360 RETURN
```

```
10 COLOR 1,15,15:SCREEN 2
20 C=ATN(1)/45
30 GOSUB 70
40 GOSUB 260
50 GOTO 50
70 FOR E=1 TO 1.25 STEP .02
100 A=22:B=A*SQR(E*E-1)
110 DRAW"BM"+STR$(128+INT(A/COS
(-80*C)))+","+STR$(95-INT(B*TAN
(-80*C)))
130 FOR TH=-80 TO 80 STEP 20
140 X=A/COS(TH*C)
150 Y=B*TAN(TH*C)
160 LINE-(128+X,95-Y),1
170 NEXT
180 DRAW"BM"+STR$(127+INT(A/COS
(100*C)))+","+STR$(95-INT(B*TAN
(100*C)))
190 FOR TH=100 TO 260 STEP 20
200 X=A/COS(TH*C)
210 Y=B*TAN(TH*C)
220 LINE-(127+X,95-Y),1
230 NEXT TH,E
250 RETURN
260 FOR E=1 TO .1 STEP -.03
270 CIRCLE(127,95),95,8,,,E
280 NEXT:RETURN
```

```
10 PMODE 3,1:PCLS2:SCREEN 1,0
20 C=ATN(1)/45
30 GOSUB 70
40 GOSUB 260
50 GOTO 50
70 FOR E=1 TO 1.25 STEP .02
100 A=22:B=A*SQR(E*E-1)
110 DRAW"BM"+STR$(128+INT(A/COS
```



A equação para um círculo também é usada para os polígonos.

```
(-80*C)))+","+STR$(95-INT(B*TAN
(-80*C)))
130 FOR TH=-80 TO 80 STEP 20
140 X=A/COS(TH*C)
150 Y=B*TAN(TH*C)
160 LINE-(128+X,95-Y),PSET
170 NEXT
180 DRAW"BM"+STR$(127+INT(A/COS
(100*C)))+","+STR$(95-INT(B*TAN
(100*C)))
190 FOR TH=100 TO 260 STEP 20
200 X=A/COS(TH*C)
210 Y=B*TAN(TH*C)
220 LINE-(127+X,95-Y),PSET
230 NEXT TH,E
250 RETURN
260 FOR E=1 TO .1 STEP .03
270 CIRCLE(127,95),95,3,E
280 NEXT:RETURN
```

Tanto a elipse quanto a hipérbole podem ser desenhadas com excentricidades diferentes — este é o papel que cabe à variável **E** ou **e**.

Para as elipses, **E** varia de 0 até 1, fazendo-as ir desde um círculo até uma reta. No programa, **E** varia de .5 a .98, o que garante que as elipses sejam suficientemente abertas.

Nas hipérboles, **E** pode variar de 1 até o infinito, mas, novamente, o programa restringe o intervalo, fazendo com que **E** varie de 1 a 2. Quanto maior for **E**, mais próxima a hipérbole estará de uma reta.

É muito fácil calcular a excentricidade de uma elipse ou de uma hipérbole. Para uma elipse de equações **X = A*COS T** e **Y = B*SIN T**, a fórmula será $E^2 = B^2/A^2 - 1$, como pode ser observado na linha 280 do programa para o Spectrum. Os outros micros não precisam calcular a variável **E**, que é simplesmente incorporada ao comando **CIRCLE** (MSX e TRS-Color) ou à simulação do mesmo (Apple, linha 280).

Para uma hipérbole cujas equações são do tipo **X = A/COS T** e **Y = B*TAN T**, temos $E^2 = 1 - B^2/A^2$. Uma versão adaptada dessa fórmula é utilizada na linha 100 do programa.

OS SEGREDOS DO SPECTRUM (1)

- COMO ESCREVER NAS LINHAS RESERVADAS DA TELA
- DELIMITAÇÃO DAS ÁREAS INDEPENDENTES DA TELA
- UMA TELA LISTRADA

Existem muitos recursos "ocultos" na ROM do ZX Spectrum. Conhecendo-os, você poderá empregar fantásticos truques de programação em BASIC, obtendo o máximo de seu computador.

O micro Sinclair ZX Spectrum ilustra muito bem o famoso ditado "tamanho não é documento". Os felizes usuários dessa popular maquininha, que já tiveram a oportunidade de conhecer mais a fundo seus recursos técnicos, podem testemunhar o quanto eles são poderosos — mais até do que os encontrados em muitos microcomputadores profissionais, maiores e mais caros.

Na série de artigos que iniciamos aqui, você verá de que modo pode explorar alguns "recursos secretos" do Spectrum e de seus compatíveis nacionais (como o TK-90X e o TK-95 da Microdigital) e internacionais.

Por que "secretos"? Como você mesmo terá oportunidade de verificar, tais recursos não estão documentados no Manual de Operação e, portanto, são desconhecidos da grande maioria dos usuários.

A TELA PROTEGIDA

É provável que você ainda não tenha descoberto a possibilidade de escrever nas duas linhas da parte de baixo da tela. Normalmente, não se usa o comando **PRINT** para escrever nessas linhas, pois elas são reservadas para a entrada de dados (**INPUT**), para a entrada e edição de programas e para a impressão de mensagens de erro do interpretador BASIC. É muito fácil, porém, modificar um comando **PRINT** de tal maneira que se possa utilizar essa área independente da tela:

```
10 PRINT 0;"Mensagem"
```

O comando funciona evidentemente dentro de um programa. O número zero, colocado entre o **PRINT** e a mensagem a ser escrita, provoca o efeito desejado. Para limpar a parte de baixo da tela sem afetar a área normal de escrita, use a instrução:

```
20 INPUT ""
```

Procure não deixar um espaço em branco entre as aspas, pois o comando não funcionará.

Experimentando um pouco mais, você fará novas descobertas sobre a parte inferior da tela. Verá, por exemplo, que ela não é limitada às duas linhas antes mencionadas. Dessa maneira, se mais palavras forem incorporadas ao programa, a parte de baixo se expandirá, "empurrando" para cima a parte superior da tela.

Pode-se também usar o comando **PRINT AT** na parte inferior da tela, neste caso, porém, a numeração deve ser independente.

Tente o seguinte:

```
10 PRINT 0;"INPUT";AT 12,0;
   "APRESENTA DICAS"
```

Se você quiser limpar a parte de baixo, após uma pequena pausa, acrescente estas linhas:

```
20 PAUSE 0
30 INPUT "":PAUSE 0
```

E, para obter sucessivas repetições, acrescente:

```
40 GOTO 10
```

Finalmente, se para você é difícil perceber onde começa cada área independente da tela, adicione ao programa a linha a seguir, e rode-o novamente. Como você verá, o comando **BORDER** coloca uma moldura de cor diferente na parte de cima da tela.

```
5 BORDER 2
```

APLICAÇÕES

Mas para que servem esses truques? Tudo depende de sua capacidade de colocar a imaginação para funcionar.

A escrita na parte inferior da tela poderia ser utilizada, entre outras coisas, para deixar fora da área principal as mensagens de prontidão para entrada de dados. Assim, ao realizar, por exemplo, uma animação gráfica, será possível colocar mensagens num ponto em que não atrapalhem o movimento nem sejam destruídas por ele.

Outra aplicação importante consiste na criação de "janelas": áreas independentes de texto ou de gráficos, muito usadas nos "pacotes" mais modernos para micros profissionais.

UMA TELA LISTRADA

Pode-se obter um outro efeito interessante de tela quando o computador está sem fazer nada — aguardando, por exemplo, que uma tecla seja pressionada (função **INKEY$**). Nesse tipo de situação, sofistique seu programa com a rotina apresentada a seguir. Ela indica o estado de espera, fazendo a moldura da tela exibir continuamente um listrado cambiante de cores:

```
1 GOTO 100
2 BORDER 1:BORDER 2:BORDER 4:
BORDER 5:BORDER 6:BORDER 7:
PAUSE 1:IF INKEY$="" THEN GOTO
10
3 LET X$=INKEY$:RETURN
80 REM ------------------
90 REM PROGRAMA PRINCIPAL
95 REM ------------------
100 GOSUB 10
110 PRINT X$:GOTO 100
```

A instrução da linha 2, **PAUSE 1**, sincroniza as mudanças nas cores da moldura de modo que ocorram simultaneamente ao início da varredura da tela. Assim, elas parecem estar estacionárias. Experimente tirar o **PAUSE 1** para ver o que acontece.

O número de comandos **BORDER** também desempenha uma função: comandos que estejam a mais ou a menos forçarão uma perda da sincronização.

Introduza a rotina da linha 2 no começo do programa principal, que deve ser colocado a partir da linha 100. A linha 1 serve apenas para pular até o início do programa. O truque é necessário, pois o **GOTO** existente na linha 2 provocará um grande retardo se a sub-rotina estiver no final da listagem. Como o interpretador BASIC procura a linha de destino a partir do início do programa, sempre que encontra um **GOTO**, a demora será maior se houver um longo programa antes da linha que contém a instrução de desvio.

UMA AGENDA ELETRÔNICA (3)

Apresentamos aqui a parte final do nosso programa de agenda e calendário. Tendo listado as últimas rotinas, você poderá, finalmente, explorar todas as possibilidades desse versátil programa.

Se você já tem as duas partes anteriores deste programa gravadas em disco ou fita, carregue-as na memória do micro e digite as linhas que faltam. O programa estará completo e você poderá começar a usá-lo para manter um registro de todos os seus compromissos financeiros, sociais ou de saúde.

É aconselhável que releia as instruções para a utilização do programa dadas nos dois primeiros artigos da série. Instruções adicionais sobre gravação e leitura de dados da fita ou disco são fornecidas a seguir.

É fácil lidar com o programa. O menu principal permite que você escolha exatamente o que deseja, em cada situação. A todo instante, você encontrará uma mensagem indicando como proceder. Todos os pontos de entrada de dados têm uma rotina de verificação. Portanto, não se preocupe muito em digitar dados incompatíveis — o programa simplesmente voltará a solicitar o dado.

GRAVAÇÃO DOS DADOS

Após entrar os dados de sua agenda, escolha a opção 8 para gravá-los. No Apple II, o programa prevê o emprego de um drive; nos demais micros, de fita cassete. Há uma rotina especial para quem utiliza o TRS-Color com drives. Se você usa fita, deixe o gravador preparado para receber os dados quando solicitar a opção 8.

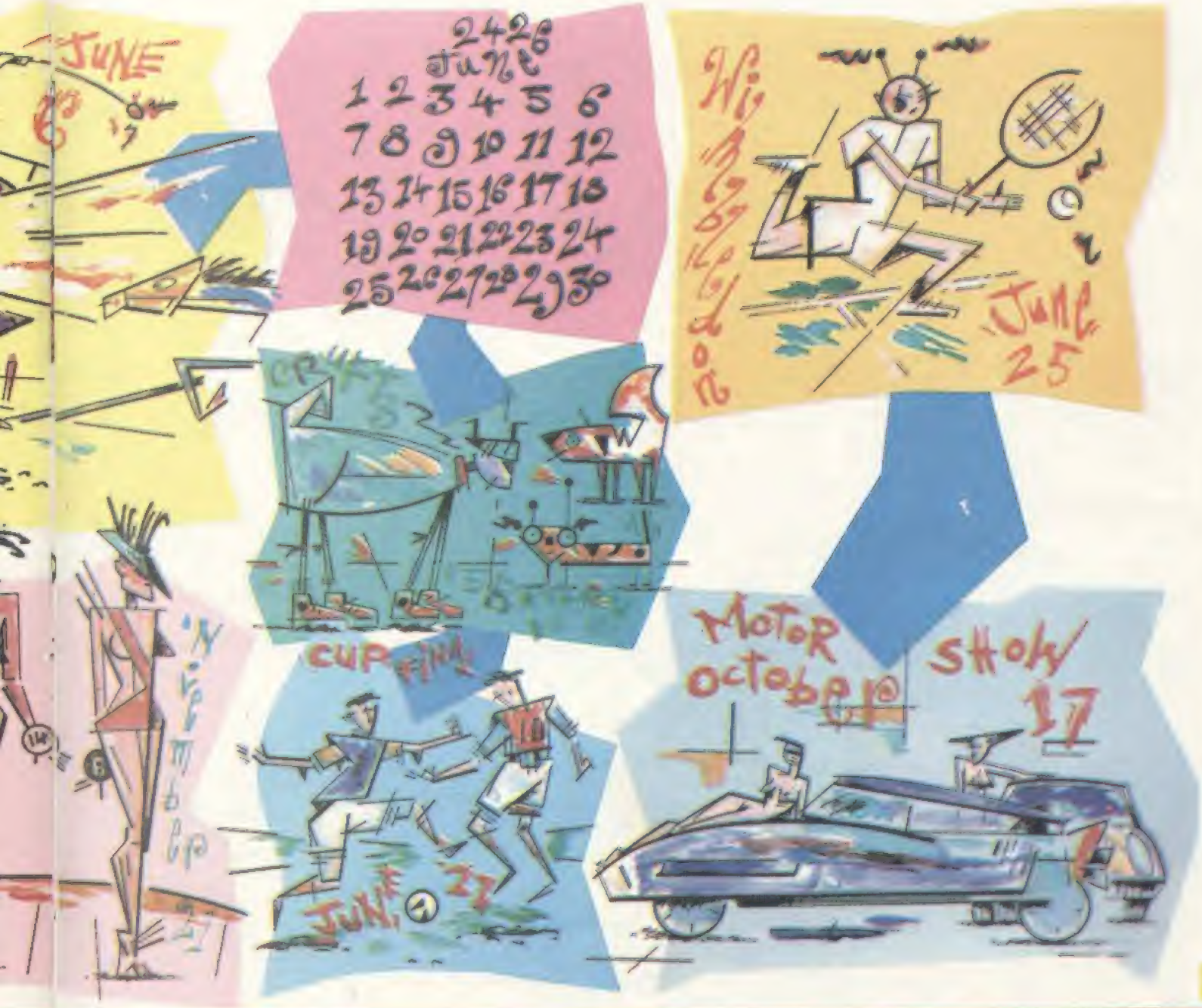
Ao usar novamente o programa, responda **S** à pergunta "Você tem listas de dados?". Os dados preexistentes serão carregados para a memória, podendo ser atualizados, apagados, listados. Antes de teclar **S**, deixe o gravador pronto para ler a fita com os dados.

O nome para esse arquivo é "DIÁRIO". Assim, adote outro nome para o programa (CALENDÁRIO, por exemplo).

```
1760 LET M4=0: LET A4=0
1770 INPUT "ANO:";YR: IF YR<175
3 OR YR>29999 THEN  GOTO 1770
1780 GOSUB 640
1790 GOSUB 2480
1800 CLS
1810 POKE 23692,255
1820 PRINT #P;"ANO ";YR
1830 PRINT #P: LET KB=0: GOSUB
1920: PRINT #P
1840 GOSUB 2460
1850 FOR z=1 TO 12
1860 LET MO=z
1870 PRINT #P;M$(MO*9-8 TO MO*9
)
1880 LET T2=5: LET S2=0: GOSUB
2020
1890 IF P=2 THEN  IF INKEY$="" 
THEN  GOTO 1890
1900 NEXT z
1910 RETURN
1920 LET X2=0: LET C2=0: LET D2
=0
1930 IF KB=0 THEN  LET X2=7: PR
INT #P
1940 IF P=3 THEN  LET KB=KB+1
1950 PRINT #P;Z$( TO X2);
1960 FOR d=1 TO 7
1970 INK 4: IF d=1 THEN  INK 2
1980 PRINT #P;Z$( TO KB);s$((d-
1)*3+1 TO (d-1)*3+3);
1990 NEXT d
2000 INK 7
2010 RETURN
2020 PRINT
2030 IF P=3 THEN  PRINT AT 10,4
; FLASH 1;"SAIDA PARA A IMPRESS
ORA"
2040 LET M5=0: LET XP=0: LET X2
=0: LET W2=0: LET A$="": LET D$
=""
2050 IF S2=1 THEN  LET A$=" ": 
LET W2=4
2060 IF S2=0 THEN  LET X2=7: LE
T W2=3
2070 IF P=3 THEN  LET A$=A$+" "
: LET W2=W2+1
2080 LET DA=1
2090 LET KB=MO: GOSUB 270: LET
M5=KB
2100 GOSUB 560: LET K2=7: LET X
P=FN M(KB)
2110 PRINT #P;Z$( TO XP*W2);
2120 LET DA=0
2130 PRINT #P;Z$( TO X2);
2140 LET DA=DA+1: LET D$=A$+(ST
R$ (DA))+" ": IF LEN D$<W2 THEN
  LET D$=D$+Z$( TO W2-LEN D$)
2150 IF A$="" THEN  PRINT #P;D$
;: GOTO 2170
2160 LET KB=T2: GOSUB 350: PRIN
T #P; INK KB;D$;
2170 LET XP=XP+1
2180 IF NOT (XP>6 OR DA=M5) THE
N  GOTO 2140
2190 LET XP=0: PRINT #P: IF S2=
1 THEN  PRINT #P
2200 IF DA<>M5 THEN  GOTO 2130
2210 IF MO=ME THEN  PRINT #P;:
PRINT #P;"Domingo de Pascoa ";M
$(ME*9-8 TO ME*9);DE
2220 IF P=3 THEN  PRINT AT 10,0
;Z$: PRINT AT 10,13;"PRONTO"
```

- ROTINAS QUE COMPLETAM O PROGRAMA
- GRAVAÇÃO DAS LISTAS DE AGENDA EM FITA
- COMO CARREGAR A INFORMAÇÃO PARA A MEMÓRIA
- ATUALIZAÇÃO DA INFORMAÇÃO
- A ROTINA DE IMPRESSÃO
- MUDANÇAS PARA SISTEMAS COM DISQUETE

```
2230 RETURN
2240 GOSUB 2510
2250 LET T2=0: LET MX=0: LET N2
=0: LET A$="": LET CL=0: LET M9
=MO: LET Y9=YR
2260 GOSUB 2480: CLS
2270 GOSUB 2570: PRINT #P
2280 PRINT #P;"DIA      ENTRADA"
: PRINT AT 0,18;"Qualquer tecla
 para"'TAB 18;"proxima entrada"
2290 PRINT #P
2300 GOSUB 2460
2310 IF MO=ME THEN  PRINT #P; I
NK 5;DE;"  Domingo de Pascoa"
2320 PRINT #P
2330 FOR t=1 TO 4
2340 LET MX=Q(t)
2350 IF MX=0 THEN  GOTO 2410
2360 FOR N=1 TO MX
2370 LET K$=L$(t,N)
2380 LET KB=t
```



```
2390 LET K2=3: GOSUB 470: IF K2
=1 THEN  PRINT #P; INK Z(t);K$(
2 TO 3);"      ";K$(10 TO )
2400 NEXT N
2410 IF INKEY$="" THEN  GOTO 24
10
2420 NEXT t
2430 FOR I=1 TO 100: NEXT I
2440 IF INKEY$="" THEN  GOTO 24
40
2450 LET MO=M9: LET YR=Y9: RETU
RN
2460 IF INKEY$="" THEN  GOTO 24
60
2470 RETURN
2480 PRINT : PRINT "DESEJA IMPR
IMIR (S/N) ?": LET K$="sn": GOS
UB 1480
2490 LET P=2: IF KB=1 THEN  LET
 P=3
2500 RETURN
2510 INPUT "MES ?";MO
2520 IF MO<1 OR MO>12 THEN  GOT
O 2510
2530 INPUT "ANO ?";YR
2540 IF YR<1735 OR YR>29999 THE
N  GOTO 2530
2550 GOSUB 640
2560 RETURN
2570 PRINT #P; PAPER 1; INK 7;M
$(MO*9-8 TO MO*9);" ";YR
2580 RETURN
```

MICRO DICAS

APLICAÇÕES PROFISSIONAIS DO PROGRAMA DE AGENDA

O programa de agenda e calendário, além de funcionar bem para a marcação e acompanhamento de compromissos e atividades de natureza pessoal, pode ter diversas aplicações profissionais, tal como a marcação de consultas em uma clínica médica ou odontológica. Para isso, será necessário modificar o programa em alguns pontos, pois o nível e o tipo de informações que uma agenda profissional requer são bem diferentes dos de uma agenda pessoal.

Tomemos como exemplo a agenda profissional de um dentista. Além de informações como data, horário da consulta, nome, sexo e idade do paciente, ela pode incorporar outros tipos de dado — entre eles, o procedimento a ser adotado (obturação, cirurgia, colocação de prótese etc.), a duração prevista para a consulta, a sala em que ela será realizada. Outros dados podem ser incluídos, como o saldo ou débito financeiro do paciente e a fonte pagadora (convênio, por exemplo).

Os relatórios obtidos a partir de uma agenda também são muito diferentes para um profissional. No caso do dentista, ele poderia precisar não só da agenda diária impressa, com a lista dos pacientes a serem atendidos e demais dados mencionados, mas, ainda, da listagem dos pacientes que faltaram, de uma análise estatística dos procedimentos realizados no mês etc.

T

```
2240 'IMPRIME MES -T2-S2
2250 M5=0:XP=0:X2=0:W2=0:A2$=""
:D2$=""
2260 IF S2=1 THEN A2$=" ":W2=4
2270 IF S2=0 THEN X2=7:W2=3
2280 IF P=2 THEN A2$=A2$+" ":W2
=W2+1
2290 DA=1
2300 KB=MO:GOSUB 230:M5=KB
2310 GOSUB 560:K2=7:XP=FNM(KB)
2320 PRINT #-P,STRING$(XP*(W2),
32);
2330 DA=0
2340 REM
2350 PRINT #-P,STRING$(X2," ");
2360 REM
2370 DA=DA+1:D2$=A2$+MID$(STR$(
DA),2)+" ":IF LEN (D2$)<W2 THEN
 D2$=D2$+" "
2380 IF A2$="" THEN PRINT #-P,D
2$;:GOTO 2400
2390 KB=T2:GOSUB 310:MID$(D2$,1
,1)=CHR$(KB):PRINT#-P,D2$;
2400 XP=XP+1
2410 IF NOT(XP>6 OR DA=M5) THEN
 2360
2420 XP=0:PRINT #-P:IF S2=1 THE
N PRINT#-P
2430 IF DA<>M5 THEN 2340
2440 IF MO=ME THEN PRINT #-P:PR
INT#-P,"DOMINGO DE PASCOA ";MID
$(MN$,ME*9-8,9);DE
2450 RETURN
2460 'ROTINA DIARIO
2470 T2=0:MX=0:N2=0:A$="":CL=0:
M9=MO:Y9=YR
2480 GOSUB 2750:GOSUB 2720:CLS
2490 GOSUB 2820:PRINT #-P
2500 PRINT#-P,"DIA      ENTRADA"
2510 GOSUB 2660
2520 IF MO=ME THEN PRINT #-P,DE
;"DOMINGO DE PASCOA":PRINT #-P
2530 FOR T2=0 TO 3
2540 CL=159+16*T2
2550 IF P=2 THEN CL=32
2560 MX=VAL(LI$(T2,0))
2570 IF MX=0 THEN 2620
2580 FOR N2=1 TO MX
2590 A$=LI$(T2,N2)
2600 KB$=A$:GOSUB 470:IF K2=1 T
HEN PRINT #-P,MID$(STR$(ASC(MID
$(A$,2,1))),2);TAB(3);CHR$(CL);
RIGHT$(A$,LEN(A$)-4)
2610 NEXT:PRINT #-P
2620 IF P=0 AND INKEY$="" THEN
2620
2630 NEXT
2640 IF INKEY$="" THEN 2640
2650 MO=M9:YR=Y9:RETURN
2660 'ESPERA POR TECLA
2670 P1=PEEK(136):P2=PEEK(137)
2680 PRINT @480,"QUALQUER TECLA
 PARA CONTINUAR";
2690 IF INKEY$="" THEN 2690
2700 PRINT @480,STRING$(30,32);
:POKE 136,P1:POKE 137,P2
2710 RETURN
2720 'IMPRESSAO
2730 PRINT:PRINT"GOSTARIA DE IM
PRIMIR (S/N) ?":KB$="SN":GOSUB
1590:IF KB=1 THEN P=2
2740 RETURN
2750 REM
2760 INPUT"MES:";MO
2770 IF MO<1 OR MO>12 THEN 2760
2780 INPUT" ANO:";YR
2790 IF YR<1753 OR YR>29999 THE
N 2780
2800 GOSUB 650
2810 RETURN
2820 REM
2830 PRINT MID$(MN$,MO*9-8,9);"
 ";YR
2840 IF P=2 THEN PRINT #-2,MID$
(MN$,MO*9-8,9);"  ";YR
2850 RETURN
```

MSX

```
2240 ' imprime mês
2250 M5=0:XP=0:X2=0:W2=0:A2$=""
:D2$=""
2260 IF S2=1 THEN A2$=" ":W2=4
2270 IF S2=0 THEN X2=7:W2=3
2280 IF P=2 THEN A2$=A2$+" ":W2
=W2+1
2290 DA=1
2300 KB=MO:GOSUB 230:M5=KB
2310 GOSUB 560:K2=7:XP=FNM(KB)
2320 PRINT#P, STRING$(XP*W2,32)
;
2330 DA=0
2340 REM
2350 PRINT#P, STRING$(X2," ");
2360 REM
2370 DA=DA+1:D2$=A2$+MID$(STR$(
DA),2)+" ":IF LEN(D2$)<W2 THEN
D2$=D2$+" "
2380 IF A2$="" THEN PRINT#P,D2$
;:GOTO 2400
2390 KB=T2:GOSUB 310:MID$(D2$,1
,1)=CHR$(KB):PRINT#P,D2$;
2400 XP=XP+1
2410 IF NOT(XP>6 OR DA=M5) THEN
 2360
2420 XP=0:PRINT#P,:IF S2=1 THEN
 PRINT#P,
2430 IF DA<>M5 THEN 2340
2440 IF MO=ME THEN PRINT#P,:PRI
NT#P,"Domingo de Páscoa:   ";DE
;"de ";MID$(MN$,ME*9-8,9)
2450 RETURN
2460 ' rotina diário
2470 T2=0:MX=0:N2=0:A$="":CL=0:
M9=MO:Y9=YR
2480 GOSUB 2750:GOSUB 2720:CLS
2490 GOSUB 2820:PRINT#P,
2500 PRINT#P,"Dia       Comprom
isso"
```

```
2510 GOSUB 2660
2520 IF MO=ME THEN PRINT#P,"dia
";DE;"Domingo de Páscoa":PRINT#
P,
2530 FOR T2=0 TO 3
2540 CL=32
2550 IF P=2 THEN CL=32
2560 MX=VAL(LI$(T2,0))
2570 IF MX=0 THEN 2620
2580 FOR N2=1 TO MX
2590 A$=LI$(T2,N2)
2600 KB$=A$:GOSUB 470:IF K2=1 T
HEN PRINT#P,MID$(STR$(ASC(MID$(
A$,2,1))),2);TAB(3);CHR$(CL);RI
GHT$(A$,LEN(A$)-4)
2610 NEXT:PRINT#P.
2620 IF P=3 AND INKEY$="" THEN
2620
2630 NEXT
2640 IF INKEY$="" THEN 2640
2650 MO=M9:YR=Y9:RETURN
2660 RETURN
2720 ' opção impressão
2730 PRINT:PRINT"Quer imprimir?
 (S/N)":KB$="SN":GOSUB 1590:IF
KB=1 THEN P=2
2740 RETURN
2750 ' rotina de data
2760 INPUT " Mês:";MO
2770 IF MO<1 OR MO>12 THEN 2760
2780 INPUT " Ano:";YR
2790 IF YR<1753 OR YR>29999 THE
N 2780
2800 GOSUB 650
2810 RETURN
2820 ' título
2830 PRINTMID$(MN$,MO*9-8,9);"
";YR
2840 IF P=2 THEN LPRINT MID$(MN
$,MO*9-8,9);"   ";YR
2850 RETURN
```

```
2240  REM  IMPRIME MES
2250 M5 = 0:XP = 0:X2 = 0:W2 =
0:A2$ = "":D2$ = ""
2260  IF S2 = 1 THEN A2$ = " ":
W2 = 4
2270  IF S2 = 0 THEN X2 = 7:W2
= 3
2280  IF P = 2 THEN A2$ = A2$ +
 " ":W2 = W2 + 1
2290 DA = 1
2300 KB = MO: GOSUB 230:M5 = KB

2310  GOSUB 560:K2 = 7:XP =  FN
 M(KB)
2320  PRINT  SPC( XP * W2)
2330 DA = 0
2340  REM
2350  PRINT  SPC( X2)
2360  REM
2370 DA = DA + 1:D2$ = A2$ +  S
TR$ (DA) + " ": IF  LEN (D2$) <
 W2 THEN D2$ = D2$ + " "
2380  IF A2$ = "" THEN  PRINT D
2$;: GOTO 2400
2390 KB = T2: GOSUB 310:D2$ =
CHR$ (KB) +  MID$ (D2$,2): PRIN
T D2$;
2400 XP = XP + 1
```

```
2410  IF  NOT (XP > 6 OR DA = M
5) THEN 2360
2420 XP = 0: PRINT : IF S2 = 1
THEN  PRINT
2430  IF DA <  > M5 THEN 2340
2440  IF MO = ME THEN  PRINT "D
OMINGO DE PASCOA:    ";DE;" DE "
; MID$ (MN$,ME * 9 - 8,9)
2450  RETURN
2460  REM  ROTINA DIARIO
2470 T2 = 0:N2 = 0:MX = 0:A$ =
"":CL = 0:M9 = MO:Y9 = YR
2480  GOSUB 2750: GOSUB 2720: H
OME
2490  PRINT D$;"PR#";P: GOSUB 2
820: PRINT
2500  PRINT "DIA       COMPROMIS
SO"
2510  GOSUB 2660
2520  IF MO = ME THEN  PRINT DE
;" - DOMINGO DE PASCOA"
2530  FOR T2 = 0 TO 3
2540 CL = 40
2550  IF P = 1 THEN CL = 32
2560 MX =  VAL (LI$(T2,0))
2570  IF MX = 0 THEN 2630
2580  FOR N2 = 1 TO MX
2590 A$ = LI$(T2,N2)
2600 KB$ = A$: GOSUB 470: IF K2
 = 1 THEN  PRINT  STR$ ( ASC (
MID$ (A$,2,1))); TAB( 3); CHR$
(32); RIGHT$ (A$, LEN (A$) - 4)

2610  NEXT : PRINT
2620  IF P = 0 AND T2 < 3 THEN
 GET R$
2630  NEXT : PRINT "<>":P = 0:
PRINT D$;"PR#";P: GET R$
2640 MO = M9:YR = Y9: RETURN
2660  RETURN
2720  REM  OPCAO IMPRESSAO
2730  PRINT : PRINT : PRINT "QU
ER UMA COPIA IMPRESSA? (S/N) ":
KB$ = "SN": GOSUB 1590: IF KB =
 1 THEN P = 1
2740  RETURN
2750  REM  ROTINA DE DADOS
2760  INPUT " MES:";MO
2770  IF MO < 1 OR MO > 12 THEN
 2760
2780  INPUT " ANO:";YR
2790  IF YR < 1753 OR YR > 2999
9 THEN 2780
2800  GOSUB 650
2810  RETURN
2820  REM  ROTINA TITULO
2830  PRINT  MID$ (MN$,MO * 9 -
 8,9);" ";YR
2840  REM
2850  RETURN
```

MUDANÇAS PARA DRIVE

Eis a rotina para adaptar o programa do TRS-Color a um drive: ela inclui apenas pequenas modificações nos comandos que lidam com o arquivo.

```
1750 OPEN"O",#1,"DIARIO/DAT"
1780 PRINT #1,LI$(N,0)
1810 FOR J=1 TO 4:PRINT#1,STR$(
ASC(MID$(LI$(N,P),J,1))):NEXT J
1820 PRINT #1,MID$(LI$(N,P),5)
1850 CLOSE #1
1890 OPEN"I",#1,"DIARIO/DAT"
1910 LINE INPUT #1,LI$(N,0)
1950 FOR J=1 TO 4:INPUT#1,NN$:L
I$(N,P)=LI$(N,P)+CHR$(VAL(NN$))
:NEXT J
1960 LINE INPUT #1,NN$:LI$(N,P)
=LI$(N,P)+NN$
1990 CLOSE #1
```

A agenda é uma espécie de banco de dados?

Sim, de certa forma, uma agenda é um banco de dados. Cada linha, reservada para a anotação de um compromisso, corresponderia a um registro de banco de dados. As informações sobre datas, horários e as descrições seriam os campos básicos.

Tanto é assim que poderíamos utilizar o programa de banco de dados publicado nos artigos das páginas 68, 688 e 706 para estruturar uma agenda diária. Com esse objetivo, definiríamos os seguintes campos:

1. DIA: dois caracteres, numérico
2. MÊS: dois caracteres, numérico
3. ANO: dois caracteres, numérico
4. HORÁRIO: quatro caracteres, numérico
5. ATIVIDADE: trinta caracteres, alfabético
6. TIPO: um caractere, alfabético
7. STATUS: um caractere, alfabético

O campo TIPO serviria para identificar a categoria de compromisso ou atividade (por exemplo, L para lazer, P para profissional, F para familiar e assim por diante). Já o campo STATUS teria a finalidade de informar se a atividade marcada foi realizada ou não.

Depois de entrar seus compromissos utilizando esse esquema, será possível pesquisar o banco de dados por meio das funções de PESQUISAR E LISTAR do programa gerenciador de bancos de dados.

Com esse procedimento, você terá transformado o programa de banco de dados não em uma simples agenda, mas num poderoso e flexível instrumento de gestão pessoal.

O JOGO A RAPOSA E OS GANSOS (1)

Era uma vez uma raposa que jogava xadrez contra um grupo de gansos e... Bem, se você quer saber o fim dessa história, leia o artigo a seguir e entre no jogo.

Um programa como o do jogo chamado *Otelo* pode ser escrito — como vimos — de maneira que o computador dispute contra um adversário humano sem que este fique numa situação de inferioridade. Mas, neste caso, com um pouco de prática, pode-se, quase sempre, ganhar do computador.

Nas próximas três seções de *Programação de Jogos*, veremos um programa mais sofisticado escrito para que o computador dispute o jogo chamado *A Raposa e os Gansos*. Esse programa oferece vários níveis de dificuldade, permitindo que o jogo se torne tão fácil ou tão difícil quanto se queira.

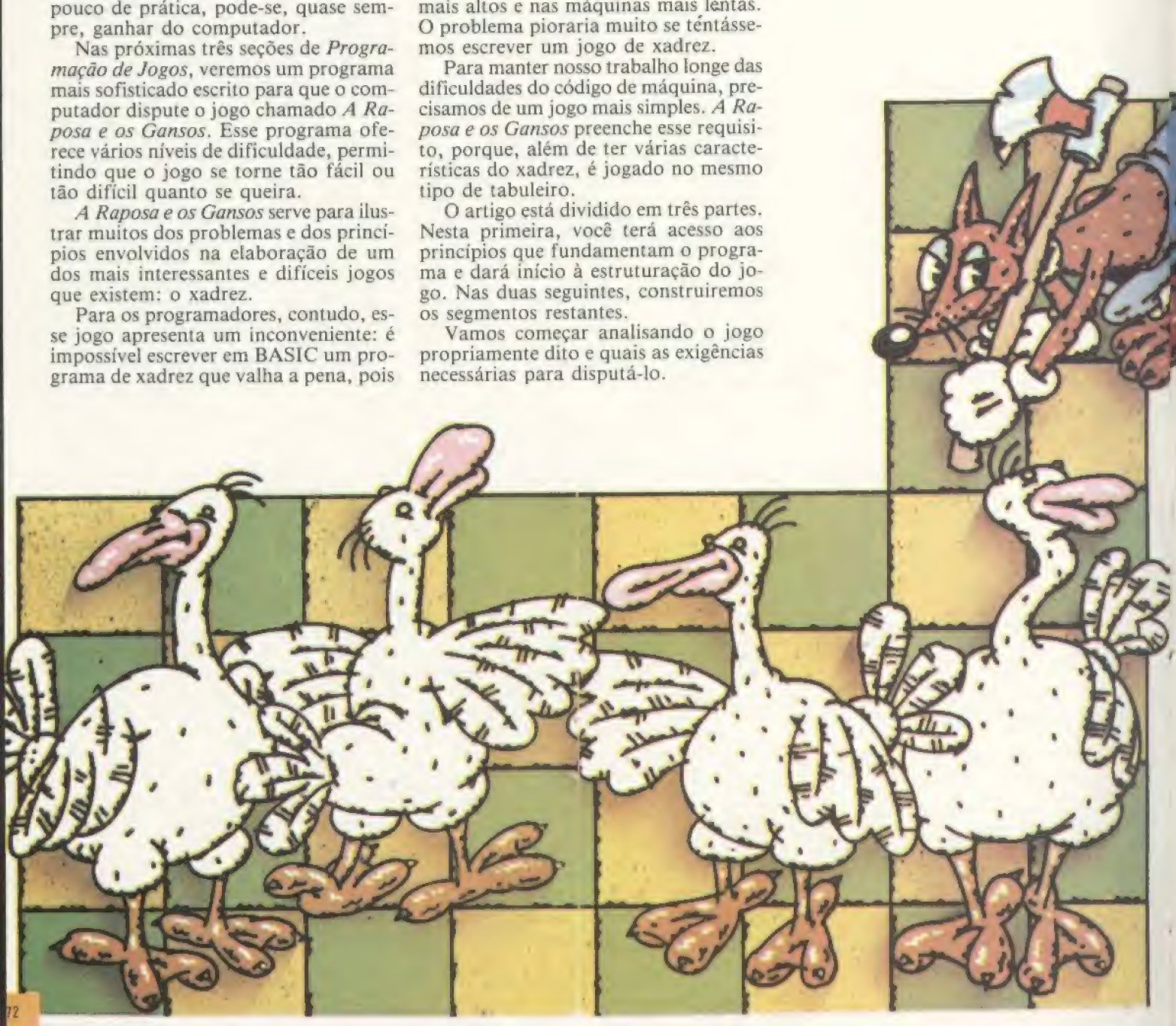
A Raposa e os Gansos serve para ilustrar muitos dos problemas e dos princípios envolvidos na elaboração de um dos mais interessantes e difíceis jogos que existem: o xadrez.

Para os programadores, contudo, esse jogo apresenta um inconveniente: é impossível escrever em BASIC um programa de xadrez que valha a pena, pois a máquina levaria longo tempo para executar cada jogada. Nos níveis superiores de *A Raposa e os Gansos*, o programa despende um bom tempo para analisar cada movimento, chegando mesmo a perder cerca de meia hora nos níveis mais altos e nas máquinas mais lentas. O problema pioraria muito se tentássemos escrever um jogo de xadrez.

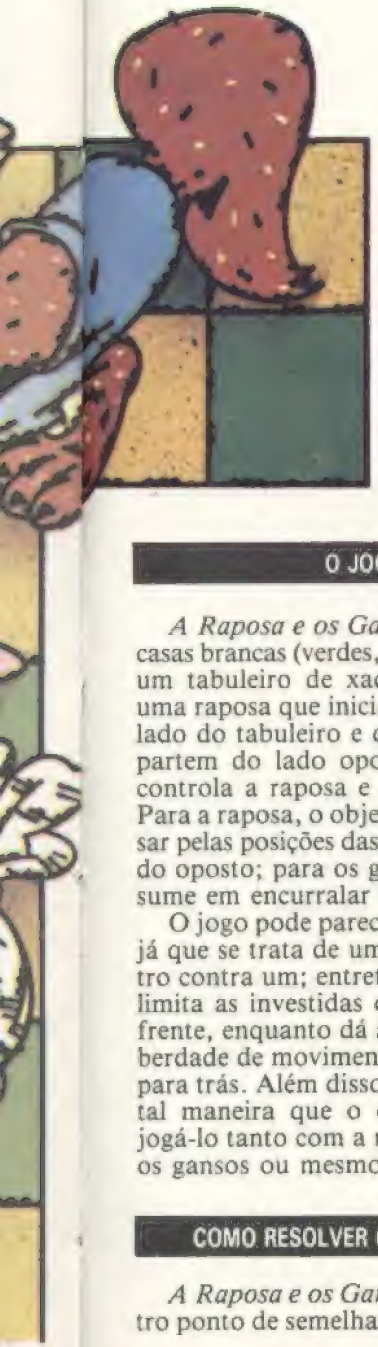
Para manter nosso trabalho longe das dificuldades do código de máquina, precisamos de um jogo mais simples. *A Raposa e os Gansos* preenche esse requisito, porque, além de ter várias características do xadrez, é jogado no mesmo tipo de tabuleiro.

O artigo está dividido em três partes. Nesta primeira, você terá acesso aos princípios que fundamentam o programa e dará início à estruturação do jogo. Nas duas seguintes, construiremos os segmentos restantes.

Vamos começar analisando o jogo propriamente dito e quais as exigências necessárias para disputá-lo.



- FAÇA DO COMPUTADOR UM JOGADOR INTELIGENTE
- RAPOSA, GANSOS E XADREZ
- TEORIA DE PROGRAMAÇÃO
- EXPLICANDO O JOGO
- COMO RESOLVER PROBLEMAS DE PROGRAMAÇÃO
- COMO AVALIAR POSIÇÕES NO JOGO
- INICIE O PROGRAMA



O JOGO

A Raposa e os Gansos é jogado nas casas brancas (verdes, nas ilustrações) de um tabuleiro de xadrez. Nele, temos uma raposa que inicia a disputa em um lado do tabuleiro e quatro gansos que partem do lado oposto. Um jogador controla a raposa e outro, os gansos. Para a raposa, o objetivo do jogo é passar pelas posições das aves e atingir o lado oposto; para os gansos, tudo se resume em encurralar a raposa.

O jogo pode parecer desequilibrado, já que se trata de uma disputa de quatro contra um; entretanto, o programa limita as investidas dos gansos para a frente, enquanto dá à raposa inteira liberdade de movimentos para a frente e para trás. Além disso, ele foi escrito de tal maneira que o computador pode jogá-lo tanto com a raposa, como com os gansos ou mesmo sozinho.

COMO RESOLVER OS PROBLEMAS

A Raposa e os Gansos apresenta outro ponto de semelhança com o xadrez: aqui também o fator sorte não é decisivo. Na verdade, o resultado depende apenas da perícia dos adversários. Embora seja teoricamente possível para o computador aprender a jogar por tentativa e erro, como um ser humano, esta não é a melhor forma de resolver os problemas envolvidos no jogo — e, de qualquer modo, um programa assim concebido não caberia no seu micro.

Programar um jogo como *A Raposa e os Gansos* ou como o xadrez é um exercício de inteligência artificial. Para constituir um desafio a um jogador humano, o computador precisa jogar inteligentemente. Porém, a máquina não tem condições de olhar para o tabuleiro e captar as relações espaciais entre as peças, tal como nós o fazemos. Assim, as posições devem ser convertidas em valores numéricos que possam ser analisados pelo computador.

Para escrever um programa que permita ao computador jogar de modo inteligente, deve-se primeiro estudar a natureza do jogo. Esta determinará o tipo de programa. É importante, também, ter como referência um método ideal, inteiramente documentado. Prestam-se bem a esse tratamento a resolução do cubo mágico, aberturas de xadrez etc. Em comparação com tais exemplos, nossa tarefa é mais simples.

Quando o elemento sorte está presente, é possível usar uma estratégia de movimento único — ou seja, o programa analisa as alternativas de jogo apenas um movimento à frente para, então, escolher a melhor jogada. O ludo, por exemplo, seria um candidato a essa estratégia, exigindo rotinas de decisão relativamente simples.

Em jogos em que a sorte conta pouco ou nada, como em *A Raposa e os Gansos*, pode-se usar uma estratégia de movimentos consecutivos, onde se analisa uma série de passos adiante para investigar os resultados possíveis. Ao encontrar o movimento mais vantajoso, o programa o executa.

AVALIAÇÃO DA POSIÇÃO

Para que o computador possa decidir qual é o melhor movimento entre vários possíveis, atribui-se um valor numérico a cada quadrado do tabuleiro. No nosso jogo, é vantajoso para a raposa estar o mais à frente possível — lembre-se de que ela ganha o jogo ao atingir o lado oposto do tabuleiro.

As colunas são numeradas alternadamente da esquerda para a direita e da direita para a esquerda. Assim, como os maiores números se alternam à direita e à esquerda, a raposa tenderá a se mover em linha reta.

Mas há outro elemento que deve ser considerado na avaliação de posições. Os cinco quadrados brancos imediatamente à frente da raposa têm um significado especial para o jogo. Se você olhar para a figura 1, verá cinco quadrados marcados de A a E. Se houver apenas um ganso em um desses quadrados, a raposa ganhará; se os gansos forem dois, mas não estiverem nas posições A ou B, ou se eles forem três, posicionados em ACD, BDE, ACE ou BCE, a vitória também caberá à raposa.

No início de cada turno, quando o computador joga tanto pela raposa quanto pelos gansos, o programa vai para uma sub-rotina que examina as posições de todas as peças e transforma essa informação em um único número, que o computador utiliza para escolher a melhor jogada. Quando a decisão é tomada, o programa converte o número novamente em posições.

BUSCA EM ÁRVORE

Os movimentos possíveis de uma posição do tabuleiro podem ser representados por uma estrutura em "árvore", com ramificações a partir da posição da peça. As alternativas de movimento a partir da posição da peça da figura 2, por exemplo, estão ilustradas no desenho da figura 3. Se examinarmos um movimento adiante, a árvore ficará mais complicada, passando a incluir o segundo nível de alternativas, também indicado na figura 3.

Observe que nem sempre teremos quatro "ramos" na estrutura, visto que a peça pode estar bloqueada por outra ou posicionada em um dos quadrados que limitam o tabuleiro.

COMO ACELERAR O PROGRAMA

Devido ao grande número de operações que devem ser feitas, o BASIC pode se revelar demasiado lento. Mas, com o uso de determinados procedimentos, é possível acelerar o programa. Inicialmente, deve-se garantir que o programa não continue avaliando o próximo passo do adversário se o seu movimento já ganhou o jogo. Isso, no entanto, só economiza tempo no fim do jogo e não durante o seu desenrolar.

Em segundo lugar, é preciso levar em conta que uma mesma configuração de posições pode ser encontrada freqüentemente, a partir de jogadas diferentes. Convém, assim, elaborar uma tabela de configurações comuns com os valores já calculados associados a elas. Essa tabela fará com que o computador pare de analisar posições que se repetem. Tenha, porém, o cuidado de não armazenar dados de uma posição cuja avaliação pode ser mais rápida do que uma consulta à tabela. Teoricamente, o emprego da tabela só é vantajoso nas situações em que o computador avalia três lances completos — ou seja, três jogadas de ambos os jogadores — ou mais. No entanto, a prática tem demonstrado que a armazenagem só vale realmente a pena quando cinco ou mais lances estiverem sendo analisados.

Finalmente, há a possibilidade de se utilizar o algoritmo alfa-beta, descoberto no início dos anos 60 por pesquisadores da área de inteligência artificial. Recorre-se a ele sempre que, numa busca em árvore, a análise envolve mais de um nível.

Voltemos à figura 3, que mostra os movimentos possíveis de uma posição da raposa. O programa vai avaliar todo o ramo A, depois o ramo B. O melhor movimento será armazenado pelo programa e comparado com os resultados de cada ramo. Ao encontrar, em qualquer ponto, um resultado pior do que um anterior, todo o ramo será rejeitado.

Quando a árvore se torna mais complexa, o algoritmo alfa-beta é realmente útil, podendo descartar cerca de 99,8% das possibilidades em um estágio bem inicial, com um ganho de tempo similar. Neste jogo, a árvore não atingirá tal nível de complexidade.

INICIE O PROGRAMA

Agora que você já tem uma noção da teoria envolvida num jogo como *A Raposa e os Gansos*, passe à digitação da primeira parte do programa. Ela se encarrega de montar alguns desenhos, mas você não os verá por enquanto, se executar o programa. Não se esqueça de gravar as linhas digitadas.

S

```
  30 DEF FN A(F)=INT (LN (F)/L2
  +.001)
 100 GOTO 2002
2002 GOSUB 5000
2006 BORDER 4: PAPER 4: CLS
2008 PRINT AT 8,7; FLASH 1;"AGU
ARDE A PREPARACAO"
5000 FOR J=USR "A" TO USR "D"+7
: READ A: POKE J,A: NEXT J: RET
URN
5070 DATA 20,28,55,127,15,20,40
,72,0,0,248,252,250,40,20,20,0,
0,0,7,205,123,60,15,12,20,31,15
2,248,216,48,224
6000 LET S$(1)="   1   2   3
4"
6010 LET S$(2)=" 8   7   6   5
 "
6015 LET S$(3)="   9  10   11
 12"
6030 LET S$(4)="16  15  14  13
 "
6040 LET S$(5)="  17  18  19  2
0"
6050 LET S$(6)="24  23  22  21
 "
6060 LET S$(7)="  25  26  27  2
8"
6070 LET S$(8)="32  31  30  29
 "
6100 RETURN
```

T

```
5 CLS:PRINT @230,"DEFININDO GRA
FICOS":GOSUB 4000
10 SCREEN 1,0:GOTO 10
4000 PMODE 3:PCLS:DIM GS(5),FX(
4),SQ(10)
4010 DRAW"BM3,0C2FGR3FR5E3RED3F
DGLGNFLNG2LH2LH":PAINT(12,5),2:
PSET(0,1,4):PSET(14,5,4):PSET(1
6,4,4):GET (0,0)-(19,9),GS,G
4020 DRAW"BM18,20C4GL13HLG2R2F2
ND4R10ND4U2":PAINT(10,22),4:PSE
T(2,21,1):GET(0,20)-(19,28),FX,
G
4030 LINE(0,0)-(175,175),PSET,B
F:COLOR 3:LINE(8,8)-(167,167),P
SET,BF
4040 FOR K=8 TO 128 STEP 40:FOR
 L=28 TO 148 STEP 40:PUT(L,K)-(
L+19,K+19),SQ,PSET:PUT(L-20,K+2
0)-(L-1,K+39),SQ,PSET:NEXT L,K
4050 PUT(68,13)-(87,21),FX,PSET
```

```
4060 FOR K=8 TO 128 STEP 40:PUT
(K,153)-(K+19,162),GS,PSET:NEXT
4070 TH$="R2ND6R2BR4D6BR4U3LBR6
ND3BU2UBF3ND3R4D3BR5U6D3NR3F3BR
4U3BU2UBF3BR2ND3R4D3BR7L3U3R3D6
L3BE3BR4RBR5RBR5R"
4080 MW$="ND6F3RU3D6BR9L4U3R4D3
BE3F3UE2BR3R3DL3D2R3BR6NU6E2F2N
U6BR3U6DJR4D3BR7L3U3R3DL3BR7NUN
R2D2BR8L3U3R3DL3BE4BR7R4D3L4D3"
4090 WG$="RD6E2F2U6BR4D6BR4U3NL
2BF3U3BU2UBF3BRNR3D3R3BR4NU6BR4
U3NL3BE3BR8L4D6R4U3BR4D3R4U3L4B
R8D3R4U3L3BR9BUL2D2R2D2L2BR9L3U
3R3DL"
4100 V$="T4O2DEFBGACDEGGDCDE"
4200 C=1:G=0:RETURN
5000 FOR K=1 TO 14:PUT(200,5)-(
210,15),SQ,PRESET:PLAY"T50AC":P
UT(200,5)-(210,15),SQ,PSET:PLAY
"DA":NEXT:RETURN
```

```
5 CLS: COLOR 1,11,11:SCREEN0
7 KEYOFF:GOSUB 4000
10 GOTO 10
4000 SCREEN2,2:FX=1:A$="":FORK=
2TO5:GS(K-1,0)=K:NEXT:FOR K=1TO
4:GS(K,1)=32-K:NEXT
4010 FOR K=1 TO 32:READ A:A$=A$
+CHR$(A):NEXT:FOR K=1 TO 4:SPRI
TE$(GS(K,0))=A$:NEXT:A$=""
4020 FOR K=1 TO 32:READ A:A$=A$
+CHR$(A):NEXT:SPRITE$(FX)=A$:
4030 LINE (10,10)-(185,185),6,B
F:LINE (18,18)-(177,177),11,BF
4050 FOR K=18 TO 138 STEP 40:FO
R L=38 TO 158 STEP 40:LINE (L,K
)-(L+19,K+19),3,BF:LINE (L-20,K
+20)-(L-1,K+39),3,BF:NEXT:NEXT
4060 PUT SPRITE FX,(78,20),6
4070 FOR K=0 TO 3:PUT SPRITE GS
(K+1,0),(18+K*40,159),15:NEXT
4080 V$="T20003DEFBGACDEGGDCDE"
4100 DATA 0,0,12,60,12,6,6,7,7,
3,3,1,0,0,1,2,0,0,0,0,0,8,8,248
,248,248,248,240,224,192,32,16
4110 DATA 0,0,0,2,2,6,62,27,3,3
,2,2,2,2,0,0,0,0,0,0,0,4,8,248,
240,240,16,16,16,16,0,0
4200 RETURN
5000 BEEP:RETURN
```

```
2  LOMEM: 16384
5  HOME : GOSUB 4000: GOSUB 310

10  END
```

```
4000  DATA  02,00,06,00,42,00
4010  DATA  45,62,63,63,46,45,4
5,53,63,63,119,41,53,127,9,45,4
5,45,62,63,63,55,45,45,53,63,63
,14,45,62,55,55,119,73,40,60,0
4020  DATA  45,45,44,44,44,54,5
5,63,62,63,46,45,45,45,45,45,45
,44,44,172,146,194,27,63,63,63,
63,55,45,45,45,45,62,63,63,63,4
6,62,62,126,73,73,37,37,36,0
4030  FOR A = 768 TO 768 + 88:
READ E: POKE A,E: NEXT
4040  POKE 233,3: POKE 232,0
4050  RETURN
5000  FOR K = 1 TO 3: PRINT  CH
R$ (7);: NEXT : RETURN
```

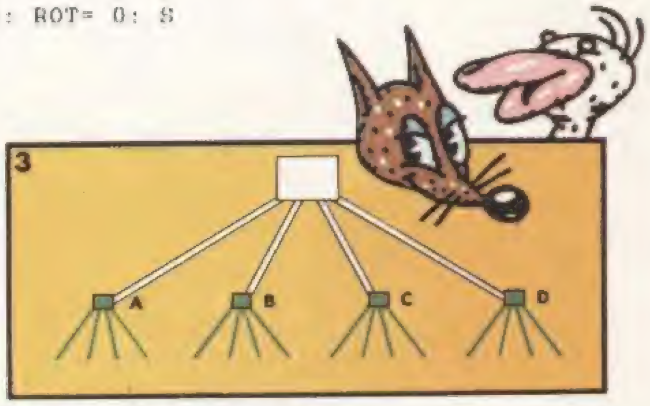
O TABULEIRO

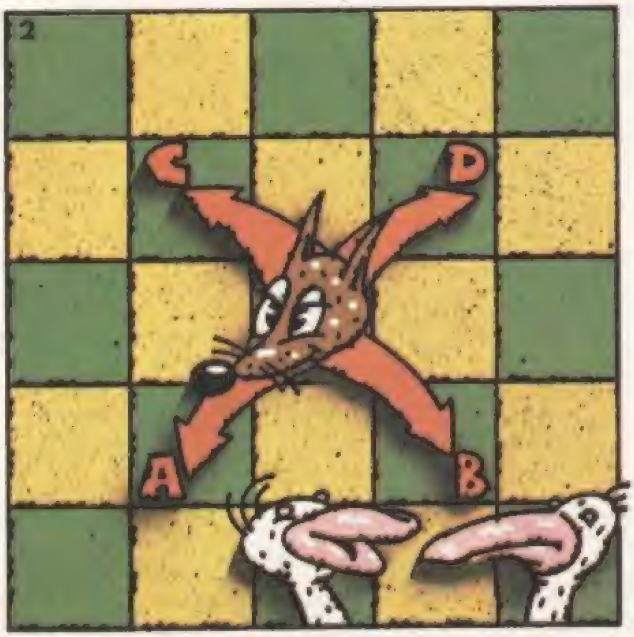
```
310 LET F=FN A(ABS (P))-30:
LET B=P/B(F): IF B<0 THEN  LET
B=B+BX: LET F=33-F
320 LET C=B/B(29): FOR A=8 TO
1 STEP -1: LET R$(A)=B$(INT (C
)+1,(2-FN W(A))): LET C=(C-INT
 (C))*16: NEXT A
330 LET R$(INT (F/4+.8))(FN C(
F)+1 TO FN C(F)+4)=F$(FN W(F/4
-.2)+1)
340 FOR A=1 TO 8: PRINT AT 2*A
,8; PAPER 7;S$(A): PRINT AT 2*
A+1,8; PAPER 7;R$(A): NEXT A:
RETURN
```

```
310  HGR : HCOLOR= 5: ROT= 0: S
CALE= 1
320 X1 = 57:X2 = 219:Y1 = 0:Y2
= 159:GS = 1:FX = 2
330  HPLOT X1,Y1 TO X2,Y1 TO X2
,Y2 TO X1,Y2 TO X1,Y1
340  FOR I = 0 TO 7
350 IN = 20 * (I / 2 <  > INT
(I / 2))
360  FOR J = 0 TO 19
370  FOR K = 0 TO 3
380  HPLOT 20 * I + J + X1 + 1,
(40 * K + IN) TO 20 * I + J + X
1 + 1,(40 * K + IN + 19)
390  NEXT : NEXT : NEXT : HCOLO
R= 3
395  DRAW FX AT X1 + 3 * 20 + 3
,Y1 + 8
400  FOR I = 0 TO 6 STEP 2: DRA
W GS AT X1 + I * 20 + 8,Y1 + 7
* 20 + 5: NEXT : RETURN
```

As linhas que vão de 310 a 350 mostram o tabuleiro com as cinco peças em posição. A sub-rotina é chamada uma vez a cada jogada, tanto pela raposa quanto pelos gansos.

As instruções para os demais micros fazem parte de uma outra sub-rotina de gráficos.





A ESCOLHA DA MEMÓRIA AUXILIAR

Existem dois tipos de memória auxiliar para micros: a fita cassete, barata e amplamente disponível, e o disquete, poderoso, mas bem mais caro. Veja aqui como fazer a melhor escolha.

Todos os computadores utilizam uma parte da memória central, chamada RAM (*Random Access Memory*), para armazenar programas e dados que são entrados pelo teclado ou carregados a partir de um periférico de armazenamento de dados. Esses dispositivos externos são necessários porque a informação armazenada na memória RAM desaparece quando o computador é desligado.

Existem diversos tipos de dispositivo de memória auxiliar destinados a microcomputadores. Os principais são as *fitas* e os *discos*. Ambos possibilitam o armazenamento permanente de informação, pois seu funcionamento é baseado no princípio da gravação magnética. Há, entretanto, uma grande diferença entre eles no que diz respeito ao custo de aquisição e à capacidade e velocidade de armazenamento.

Os discos, que constituem os dispositivos mais sofisticados e caros da memória auxiliar, dotam o computador pessoal de poderosos recursos, permitindo a programação de aplicações mais complexas, como bancos de dados de uso profissional e sistemas de controle para pequenas empresas.

ARMAZENAMENTO EM FITA

O periférico de armazenamento de dados mais utilizado em computadores pessoais é o gravador comum de fita cassete, que alia conveniência operacional e baixo custo. Para a maioria dos usuários domésticos, este é o único fator que conta.

O gravadores de audiocassete existentes no mercado são, quase todos, bastante adequados para essa finalidade. A vantagem adicional é que existe uma enorme quantidade de software de preço acessível disponível em fita cassete, para a maioria dos computadores pessoais (sobretudo na área de jogos e entretenimento).

A tecnologia dos gravadores-cassete é muito simples. Seu uso com computadores requer apenas que um sinal audível seja gerado pela interface adequada para a gravação na fita. Assim, para que o sistema funcione a contento, basta montar uma conexão entre o computador e o gravador e ajustar corretamente os níveis de volume e tonalidade. Com alguns cuidados periódicos de manutenção, o usuário não terá problemas por longo tempo.

Bem, isto é o que acontece na teoria! Na prática, contudo, não é tão fácil conseguir a combinação ideal entre o periférico e o micro.

QUALIDADE DE GRAVAÇÃO

Uma boa maneira de contornar dificuldades mais sérias consiste em comprar um gravador apropriado para microcomputadores (*data recorder*). Eles diferem dos gravadores comuns por disporem de circuitos eletrônicos internos que processam sinais mais "limpos" de transmissão digital.

Gravadores especiais, como, por exemplo, os destinados a computadores da linha MSX, dão menos problemas na gravação e leitura de dados digitais e, também, possibilitam uma conexão mais simples com o computador. Nem esses aparelhos, porém, estão totalmente livres de falhas, pois é o sistema de gravação em fita, em si, que apresenta certas desvantagens técnicas.

O principal ao se usar fita é aumentar ao máximo a qualidade de gravação: esta é uma área, portanto, sobre a qual o usuário pode exercer algum controle. Convém lembrar que, mesmo quando a gravação foi bem feita, o sinal deteriora-se ao ser lido por um gravador de baixa qualidade. O resultado é pior ainda quando se tenta gravar alguma coisa com um aparelho que esteja em más condições.

A qualidade da fita também é muito importante. Basicamente, deve-se evitar a tentação de utilizar uma fita velha, sobretudo se for de longa duração (C-90 ou C-120).

Há um conflito aqui, pois, como a

- ARMAZENAMENTO EM FITA
- QUALIDADE DA GRAVAÇÃO
- TAXA DE TRANSMISSÃO
- ALÇAS DE FITA
- ACESSO SERIAL
- ACIONADORES DE DISCOS
- O SISTEMA OPERACIONAL DE DISCOS
- INTERFACES
- TIPOS DE SOFTWARE

capacidade da memória é limitada pelo método de gravação empregado, seria desejável ampliá-la, usando-se fitas de maior duração. Estas, porém, são bem mais finas do que as C-30 e, em conseqüência, podem sofrer alongamentos por efeito do vaivém contínuo e pela força relativamente grande exercida pelo rebobinamento e avanço rápidos.

O alongamento determina perda de bytes na fita e variações na amplitude do sinal. Ambos os efeitos são desastrosos, no que diz respeito às características de leitura por computadores: um byte perdido invalida a leitura de todo um programa, por exemplo. Na melhor das hipóteses, isto pode levar a carregamentos irregulares. Na pior das hipóteses, dados são perdidos durante a gravação!

Uma fita C-30 é a melhor opção prática: ela suporta quinze minutos de gravação de cada lado, o que é mais do que suficiente para armazenar programas muito extensos (até 64 Kbytes) na maioria dos computadores. Alguns micros das linhas Sinclair e TRS-80, entretanto, têm baixas velocidades de transmissão (300 baud), podendo exceder a capacidade da fita para dados e programas. Nesse caso, recomenda-se usar fitas C-60 de boa qualidade, mas sempre com muito cuidado.

Um programa de 32 Kbytes, por exemplo, gasta somente três a quatro minutos de fita nos micros das linhas TRS-Color e MSX, cerca de seis minutos nos micros da linha Spectrum, e quase quinze minutos no ZX-81. Faz mais sentido, portanto, recorrer a fitas de curta duração, gravando-se apenas um ou dois programas de cada lado, com uma duplicata, para fins de segurança. Esse procedimento não só favorece a conservação da fita como também torna muito mais fácil verificar onde está o início de uma gravação.

Existem fitas cassete de alta qualidade, feitas especialmente para gravação de dados digitais. Elas são normalmente disponíveis em três durações: C10, C15 e C20. É sempre bom testá-las antes de comprá-las em grande quantidade, pois são mais caras do que as fitas comuns.

TAXA DE TRANSMISSÃO

Os problemas com fitas cassete ficam bem mais sérios quando a taxa de transmissão de dados entre gravador e computador é mais alta. A taxa de transmissão, ou seja, a velocidade com que os bytes passam do computador para o gravador e vice-versa, é medida em bauds. Um baud, a grosso modo, corresponde ao número de bits transmitidos por segundo (na realidade, o número é calculado com base em um esquema um pouco mais complicado).

Quanto mais alta for a taxa de transmissão utilizada pelo computador, melhor deverá ser a qualidade do gravador e da fita cassete. Uma fita pode funcionar perfeitamente bem com um ZX-81 ou TRS-80 Modelo I, mas falhará se usada com um TK-85 com função *speed*, ou com um MSX. O mesmo problema ocorrerá até com computadores mais lentos, em determinadas situações — por exemplo, na gravação de jogos com rotinas próprias de carregamento rápido.

ALÇAS DE FITA

Para evitar os transtornos causados por gravadores convencionais de fita cassete, alguns fabricantes desenvolveram um sistema diferente: o cassete de fita sem fim (alça de fita). Os mais conhecidos são o *Microdrive* e o *Wafadri-*

ve, para os micros da linha Sinclair Spectrum, mas existem diversas unidades disponíveis para computadores de outras marcas, usando interfaces padronizadas, como o RS-232C. Dispositivos desse tipo, infelizmente, não são encontrados no Brasil.

O acionador de alça de fita (*tape loop*) é um gravador que usa um cassete especial, com uma bobina sem fim (em alça). A fita desloca-se apenas em uma direção, mas muito rapidamente, de maneira que o ponto de início de um programa é atingido sem que seja necessário rebobinar a fita. Além de economizar tempo, o processo também protege a fita do desgaste provocado pelo avanço e rebobinamento rápidos dos gravadores convencionais.

Um Microdrive, por exemplo, com sua velocidade e capacidade (cerca de 100 Kbytes), permite que se eliminem todos os problemas das fitas convencionais. Seu desempenho, em alguns casos, aproxima-se do de acionadores de disquetes, muito maiores e mais caros.

Há algumas desvantagens, entretanto, nesse tipo de dispositivo: os cartuchos de fita são bem mais caros, a oferta de software no formato adequado é menor e, mesmo com maior velocidade e capacidade, ele não se presta para muitas aplicações nos campos educacional e profissional.

ACESSO SERIAL

Em qualquer sistema que utiliza fita, a transferência de dados entre gravador e computador é realizada sob a forma de uma seqüência contínua de bytes, ou, em outras palavras, como uma fila indiana, do primeiro ao último byte. Para levar os dados ou programas de volta à memória, é preciso achar o ponto de início, e carregar a partir daí. Esse tipo de processo é chamado de *acesso serial*.

Para aplicações mais simples, como jogos ou outros programas pequenos, o método não apresenta desvantagens sérias, pois o programa precisa ser localizado e carregado apenas uma vez. Entretanto, oferece inconvenientes para aplicações em que, com freqüência, o computador precisa carregar dados ou segmentos de programa, em ordem diferente daquela em que estão gravados na fita. É o que ocorre com as aplicações de bancos de dados. Um dispositivo de alça de fita pode, quando o volume de dados é pequeno, "simular" um disquete, através da marcação de "blocos" fixos de dados na fita e da localização rápida dos mesmos por um software especial. Porém, havendo um grande volume de dados, a única solução é o acionador de disquetes.

ACIONADORES DE DISCOS

Os acionadores de discos são os dispositivos mais rápidos de memória auxiliar para microcomputadores. O tempo médio necessário para a localização de blocos de dados no disquete é medido, muitas vezes, em milésimos de segundo. A rapidez da gravação e a leitura de programas torna o dispositivo atraente principalmente para quem não consegue suportar a lentidão do **CSAVE** e **CLOAD** dos gravadores.

Os discos magnéticos são, além disso, muito mais confiáveis e seguros do que as fitas, desde que se tomem algumas precauções elementares. A perda de bytes por falhas no sistema de leitura ou gravação é rara e, quando ocorre, nem sempre o arquivo inteiro ou o disco ficam inutilizados.

Entretanto, o uso de discos impõe uma nova série de tarefas e procedimentos básicos, que exigem um grau de meticulosidade e precisão desnecessário com as fitas. Se você não tomar cuidado e utilizar inadequadamente os comandos de manipulação de arquivos em discos, disponíveis nos sistemas operacionais, as conseqüências poderão ser desastrosas. Existe, por exemplo, um comando que apaga um disco inteiro, em um piscar de olhos.

A fantástica capacidade de armazenamento dos discos magnéticos é uma forte razão para que se redobrem as precauções. Existem modelos com 10 a 20 megabytes de capacidade total, ou seja, aptos a armazenar até vinte milhões de caracteres (o que equivale, mais ou menos, à quantidade de texto de uma enciclopédia de trinta volumes!). Mais comumente, entretanto, a capacidade dos disquetes disponíveis para computadores pessoais é bem mais modesta: algo entre 150 a 360 Kbytes. Mas, mesmo assim, corresponde a uma grande quantidade de dados valiosos.

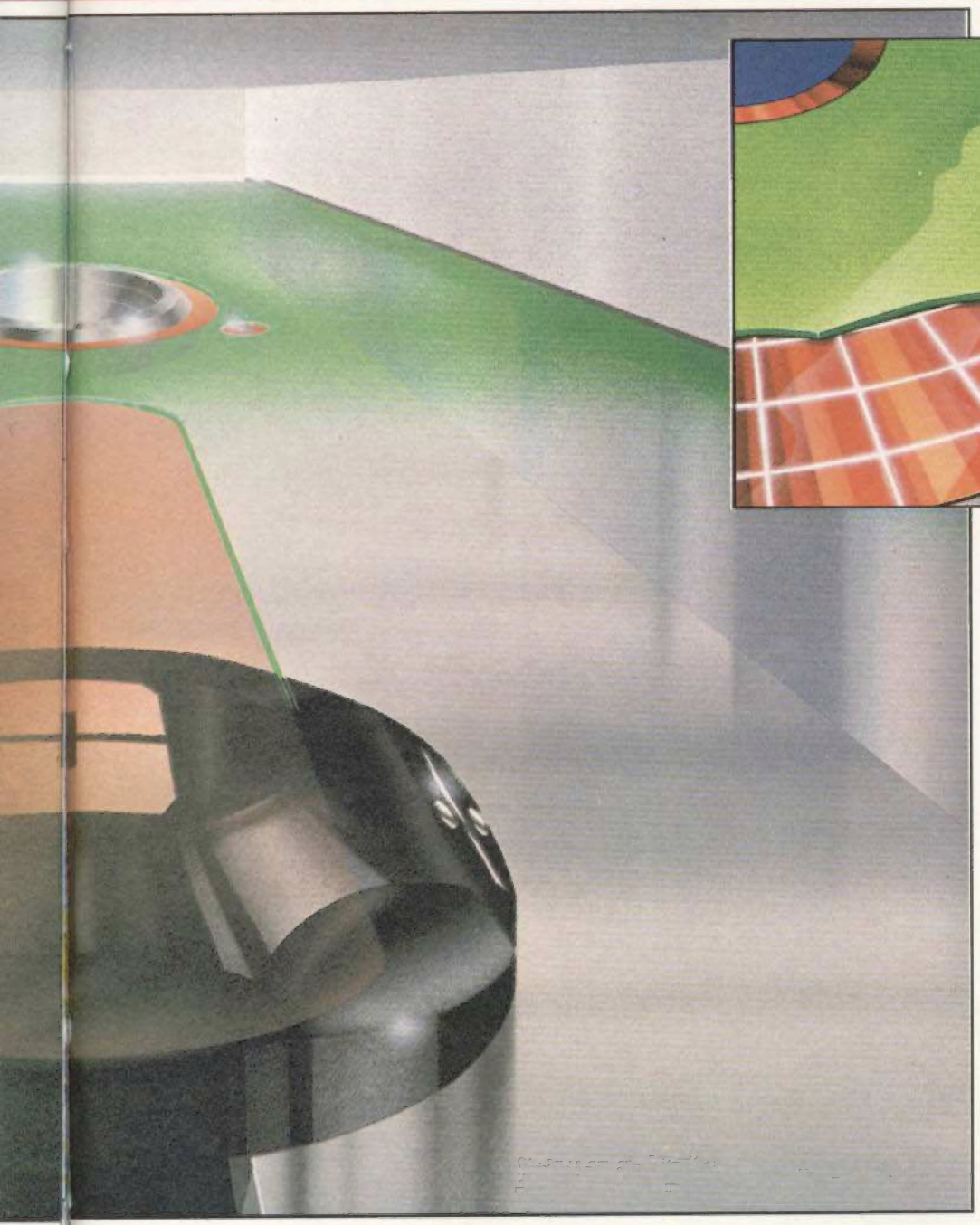
Infelizmente, os disquetes são vulneráveis a dobras, a arranhões e a agressões físicas do ambiente, tais como umidade e sujeira. Além disso, são sensíveis a campos magnéticos causados por altofalantes, aparelhos de TV, motores de geladeira etc.

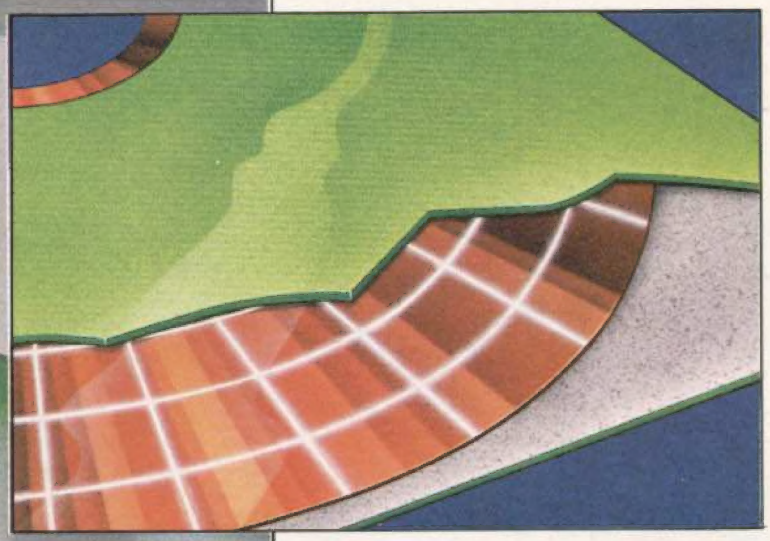
Para armazenamento de longa duração, a fita constitui um candidato bem melhor do que o disco. Assim, grandes empresas costumam manter seus arquivos vitais em bobinas de fitas, confiando aos discos apenas as operações diárias, mais rápidas.

Por todas as razões expostas, recomenda-se aos usuários de discos que façam, regularmente, cópias de segurança (*back-up*) dos arquivos de dados e de programas. A freqüência dessa operação depende muito da quantidade de informações geradas ou alteradas.

COMO FUNCIONA UM ACIONADOR

O funcionamento de um acionador de discos não é muito diferente do funcionamento de um gravador de fitas. O





O disco é formatado antes de ser usado, ou seja, é dividido em trilhas e setores. Cada setor tem a capacidade de 256 bytes.

sistema de leitura e gravação trabalha sobre um disco, chato, recoberto de material ferromagnético, como a fita, só que obedecendo a padrões de qualidade de muito mais rigorosos.

Como o gravador, o acionador de discos possui uma cabeça de leitura e gravação, que é mantida em contato com a superfície magnética. A diferença é que a cabeça também pode ser movimentada, pulando de trilha para trilha da mesma maneira que o braço de um fonógrafo. Assim, com o disco rolando em alta velocidade, atinge-se rapidamente qualquer um de seus pontos.

Antes de sua utilização, o disco é *formatado*, ou seja, sua área total é dividida em *trilhas* concêntricas e estas, por sua vez, em segmentos iguais, chamados de setores. Os setores se subdividem em *blocos*, cada qual capaz de armazenar entre 128 a 256 bytes de informação. Esses blocos podem ser imaginados como se fossem as "gavetas" de um escaninho.

Um programa chamado *sistema operacional de discos* (DOS) possibilita a gravação ou leitura de informações nos blocos (cada um tem um endereço distinto, como na RAM). O DOS é um requisito essencial para a operação de um acionador de discos.

A cabeça de leitura e gravação pode ser direcionada pelo DOS, de modo a se posicionar sobre qualquer setor na superfície do disco, com uma velocidade bastante grande. Por isso, o disco magnético é um dispositivo de acesso arbitrário, muitas vezes mais rápido do que o acesso serial ou seqüencial da fita. Ocorre que ele também é um dispositivo de acesso serial, só que limitado a um setor por vez. Isso torna o sistema muito mais veloz, flexível e poderoso do que o baseado em fita.

TIPOS DE DISCO

Os discos magnéticos para computadores dividem-se, basicamente, em dois tipos: os *discos rígidos* e os *discos flexíveis*.

O disco rígido, de preço bem mais elevado do que o flexível, é confeccionado em metal, e, em geral, não é intercambiável, ou seja, permanece sempre dentro do acionador. A cabeça não encosta em sua superfície, mas paira a alguns milésimos de milímetro da mesma.

Graças a essa característica, o desgaste desse disco é nulo, e a densidade da gravação, muito maior. Atualmente, os discos rígidos para micro — também chamados discos *Winchester* — têm capacidades entre 5 a 40 Mbytes, dependendo do modelo.

O disco flexível, genericamente denominado disquete (ou *floppy*), é feito de plástico e protegido dentro de um envelope semi-rígido, lubrificado internamente. Pode ser adquirido em três tamanhos: 8 polegadas (19 cm), 5,25 po-

legadas (12,5 cm) ou 3,5 polegadas (8,5 cm) de diâmetro. O mais utilizado em computadores pessoais ainda é o de 5,25 polegadas, chamado de minidisquete, embora os microdisquetes (3,5 polegadas) estejam se tornando cada vez mais populares. Estes são embalados em cartuchos rígidos, o que lhes confere vantagens marcantes quanto ao manuseio e segurança. Sua capacidade é também bem maior, devido à superioridade na densidade de gravação.

A velocidade de rotação de um disco, seu tamanho, tipo de superfície e a presença ou não de uma segunda cabeça do outro lado do disco condicionam a capacidade e velocidade de acesso dos diversos modelos existentes. O usuário deve, portanto, levar em conta todos esses fatores, se pretende escolher o acionador de discos mais adequado às suas necessidades.

Os discos magnéticos classificam-se, ainda, em duas outras categorias: a categoria dos que só podem ser gravados em um lado (*discos de face simples*) e a dos que permitem a gravação nos dois lados, simultaneamente (*discos de face dupla*).

A *densidade de gravação* também diferencia os discos. Os mais baratos e de menor qualidade têm *densidade simples*: sua capacidade total é bastante restrita. Os de maior capacidade e velocidade de transmissão têm *densidade dupla*. Finalmente, existem os discos de *densidade quádrupla*, com o dobro do número de trilhas dos outros discos. Estes utilizam uma técnica de pulsos de gravação diferente da técnica dos demais e oferecem o grau máximo de capacidade e velocidade. Naturalmente, o preço do disquete aumenta de acordo com sua capacidade.

Para maior segurança, devem ser usados os disquetes com a especificação correta para cada densidade. Mas muitos usuários que trabalham com acionadores de face simples (linha Apple, por exemplo) costumam utilizar também o outro lado disponível no disquete, fazendo um pequeno orifício quadrado na margem não perfurada do envelope. Tal procedimento deve ser evitado no caso de discos especificados para densidade simples. Além disso, convém levar em conta que detritos oriundos da perfuração podem danificar o disco e, até mesmo, a cabeça do acionador.

SISTEMAS OPERACIONAIS

O controle dos acionadores de discos exige um software especial. Este é fornecido junto com o acionador, formando com ele uma unidade inseparável na utilização por um computador. Essencialmente um sistema operacional, o software gerencia todos os processos de localização de informação e de transferência entre computador e disco.

Em geral, o sistema operacional de disco (DOS) não reside na memória ROM permanente do computador, precisando ser carregado na RAM a partir do próprio disco, antes de ser ativado. Assim, uma parte da RAM é utilizada, ficando menos disponível para os programas e dados do usuário.

Alguns computadores pessoais podem usar mais de um sistema operacional com o mesmo acionador de disquetes. Certos sistemas operacionais, por outro lado, têm mais comandos ou, ainda, são mais poderosos e flexíveis do que outros. Mas é preciso cautela na escolha, pois a formatação imposta ao disco muitas vezes varia entre sistemas operacionais. Em conseqüência, o usuário não conseguirá utilizar programas feitos para um DOS, se tiver carregado um DOS diferente no computador. Na maioria das vezes, porém, não há sequer a possibilidade de escolha entre vários tipos de DOS, pois só existe um para aquela marca de computador.

INTERFACES E CONEXÕES

Se você tem a intenção de adquirir um ou mais acionadores de discos, considere cuidadosamente se a unidade em vista é realmente compatível com sua máquina. Além de um DOS adequado, é necessário também escolher um disco com interface que se adapte a seu sistema. Consulte os fabricantes — do computador e da unidade de discos — e procure aconselhar-se junto a colegas e conhecidos que já tenham tido sucesso na compra de uma unidade de disco para um computador igual ao seu.

SOFTWARE PARA DISCO

Comprar um acionador de discos pode ser muito bom para incrementar seu sistema, mas, para uma boa escolha, convém ter claro o que você pretende fazer com ele. Quer esteja pensando em desenvolver seus próprios programas — em Assembler, BASIC, PASCAL ou outra linguagem de programação —, quer pretenda apenas utilizar "pacotes" prontos de software, verifique primeiro se as ferramentas de programação ou os próprios programas estão disponíveis em quantidade e qualidade para o sistema que você tem em mente.

Sistemas baseados em microdisquetes, por exemplo, são recentes no mercado, mesmo em termos internacionais, e ainda existe muito pouco software para esse formato. Também é pequena a oferta de software próprio para sistemas de disquetes para a linha MSX. À medida que um sistema vai se difundindo, porém, essa situação tende a mudar. Mas micros implantados há mais tempo no mercado — como o Apple e o TRS-80 — terão sempre maior variedade de software que os demais.

Existe ainda a alternativa de se transferir software de um formato para outro. Certos programas comercialmente disponíveis, por exemplo, tornam possível copiar software de fita para disco; paralelamente, alguns sistemas operacionais reconhecem discos de diferentes densidades, e assim por diante. Em um próximo artigo trataremos em detalhes desse assunto.

COMO ESCOLHER?

O custo é, certamente, um fator bastante significativo no momento de decidir entre um sistema baseado em fita e outro baseado em disco. Muitas das vantagens dos discos não são importantes para usuários domésticos interessados apenas em aplicações como jogos. Para a maioria dos computadores pessoais não existem, em disquete, muitas opções de programas recreativos ou de videogames, e o pouco que se encontra custa mais caro do que a mesma versão para fita. Em contrapartida, há a possibilidade de que a versão para disquete seja mais sofisticada.

As unidades de disco são muito mais caras do que as de fita, custando mais do que o próprio computador. Às vezes, o preço não inclui o cabo de ligação, que também é caro.

É preciso considerar, ainda, o custo do suporte de informação. O preço de um disquete é muitíssimo mais elevado do que o de uma fita cassete de boa qualidade. Normalmente um disquete de densidade quádrupla custa quatro ou cinco vezes mais do que um de densidade simples. No entanto, deve-se levar em conta que, ao contrário da fita, o disquete permite aproveitamento *total* de sua capacidade.

Já os discos rígidos são tremendamente caros e úteis apenas em aplicações mais profissionais. É claro que seria muito bom ter todo o seu software disponível instantaneamente no mesmo disco, e carregá-lo em um piscar de olhos. Mas será que isso vale o preço de um Winchester?

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR LINHA

LINHA	FABRICANTE	MODELO
Apple II +	Appletronica	Thor 2010
Apple II +	CCE	MC-4000 Exato
Apple II +	CPA	Absolutus
Apple II +	CPA	Polaris
Apple II +	Digitus	DGT-AP
Apple II +	Dismac	D-8100
Apple II +	ENIAC	ENIAC II
Apple II +	Franklin	Franklin
Apple II +	Houston	Houston AP
Apple II +	Magnex	DM II
Apple II +	Maxitronica	MX-2001
Apple II +	Maxitronica	MX-48
Apple II +	Maxitronica	MX-64
Apple II +	Maxitronica	Maxitronic I
Apple II +	Microcraft	Craf II Plus
Apple II +	Milmar	Apple II Plus
Apple II +	Milmar	Apple Master
Apple II +	Milmar	Apple Senior
Apple II +	Omega	MC-400
Apple II +	Polymax	Maxxi
Apple II +	Polymax	Poly Plus
Apple II +	Spectrum	Microengenho I
Apple II +	Spectrum	Spectrum ed
Apple II +	Suporte	Venus II
Apple II +	Sycomig	SIC I
Apple II +	Unitron	AP II
Apple II +	Victor do Brasil	Elppa II Plus
Apple II +	Victor do Brasil	Elppa Jr.
Apple IIe	Microcraft	Craft IIe
Apple IIe	Microdigital	TK-3000 IIe
Apple IIe	Spectrum	Microengenho II
MSX	Gradiente	Expert GPC-1
MSX	Sharp	Hotbit HB-8000
Sinclair Spectrum	Microdigital	TK-90X
Sinclair Spectrum	Timex	Timex 2000
Sinclair ZX-81	Apply	Apply 300
Sinclair ZX-81	Engebras	AS-1000
Sinclair ZX-81	Filcres	NEZ-8000
Sinclair ZX-81	Microdigital	TK-82C
Sinclair ZX-81	Microdigital	TK-83
Sinclair ZX-81	Microdigital	TK-85
Sinclair ZX-81	Prologica	CP-200
Sinclair ZX-81	Ritas	Ringo R-470
Sinclair ZX-81	Timex	Timex 1000
Sinclair ZX-81	Timex	Timex 1500
TRS-80 Mod. I	Dismac	D-8000
TRS-80 Mod. I	Dismac	D-8001/2
TRS-80 Mod. I	LNW	LNW-80
TRS-80 Mod. I	Video Genie	Video Genie I
TRS-80 Mod.III	Digitus	DGT-100
TRS-80 Mod.III	Digitus	DGT-1000
TRS-80 Mod.III	Kemitron	Naja 800
TRS-80 Mod.III	Prologica	CP-300
TRS-80 Mod.III	Prologica	CP-500
TRS-80 Mod.III	Sysdata	Sysdata III
TRS-80 Mod.III	Sysdata	Sysdata Jr.
TRS-80 Mod.IV	Multix	MX-Compacto
TRS-80 Mod.IV	Sysdata	Sysdata IV
TRS-Color	Codimex	CS-6508
TRS-Color	Dynacom	MX-1600
TRS-Color	LZ	Color 64
TRS-Color	Microdigital	TKS-800
TRS-Color	Prologica	CP-400

CLASSIFICAÇÃO DOS MICROCOMPUTADORES POR NOME

FABRICANTE	MODELO	PAÍS	LINHA
Appletronica	Thor 2010	Brasil	Apple II +
Apply	Apply 300	Brasil	Sinclair ZX-81
CCE	MC-4000 Exato	Brasil	Apple II +
CPA	Absolutus	Brasil	Apple II +
CPA	Polaris	Brasil	Apple II +
Codimex	CS-6508	Brasil	TRS-Color
Digitus	DGT-100	Brasil	TRS-80 Mod.III
Digitus	DGT-1000	Brasil	TRS-80 Mod.III
Digitus	DGT-AP	Brasil	Apple II +
Dismac	D-8000	Brasil	TRS-80 Mod. I
Dismac	D-8001/2	Brasil	TRS-80 Mod. I
Dismac	D-8100	Brasil	Apple II +
Dynacom	MX-1600	Brasil	TRS-Color
ENIAC	ENIAC II	Brasil	Apple II +
Engebras	AS-1000	Brasil	Sinclair ZX-81
Filcres	NEZ-8000	Brasil	Sinclair ZX-81
Franklin	Franklin	USA	Apple II +
Gradiente	Expert GPC1	Brasil	MSX
Houston	Houston AP	Brasil	Apple II +
Kemitron	Naja 800	Brasil	TRS-80 Mod.III
LNW	LNW-80	USA	TRS-80 Mod. I
LZ	Color 64	Brasil	TRS-Color
Magnex	DM II	Brasil	Apple II +
Maxitronica	MX-2001	Brasil	Apple II +
Maxitronica	MX-48	Brasil	Apple II +
Maxitronica	MX-64	Brasil	Apple II +
Maxitronica	Maxitronic I	Brasil	Apple II +
Microcraft	Craft II Plus	Brasil	Apple II +
Microcraft	Caft IIe	Brasil	Apple IIe
Microdigital	TK-3000 IIe	Brasil	Apple IIe
Microdigital	TK-82C	Brasil	Sinclair ZX-81
Microdigital	TK-83	Brasil	Sinclair ZX-81
Microdigital	TK-85	Brasil	Sinclair ZX-81
Microdigital	TK-90X	Brasil	Sinclair Spectrum
Microdigital	TKS-800	Brasil	TRS-Color
Milmar	Apple II Plus	Brasil	Apple II +
Milmar	Apple Master	Brasil	Apple II +
Milmar	Apple Senior	Brasil	Apple II +
Multix	MX-Compacto	Brasil	TRS-80 Mod.IV
Omega	MC-400	Brasil	Apple II +
Polymax	Maxxi	Brasil	Apple II +
Polymax	Poly Plus	Brasil	Apple II +
Prologica	CP-200	Brasil	Sinclair ZX-81
Prologica	CP-300	Brasil	TRS-80 Mod.III
Prologica	CP-400	Brasil	TRS-Color
Prologica	CP-500	Brasil	TRS-80 Mod.III
Ritas	Ringo R-470	Brasil	Sinclair ZX-81
Sharp	Hotbit HB-8000	Brasil	MSX
Spectrum	Microengenho I	Brasil	Apple II +
Spectrum	Microengenho II	Brasil	Apple IIe
Spectrum	Spectrum ed	Brasil	Apple II +
Suporte	Venus II	Brasil	Apple II +
Sycomig	SIC I	Brasil	Apple II +
Sysdata	Sysdata III	Brasil	TRS-80 Mod.III
Sysdata	Sysdata IV	Brasil	TRS-80 Mod.IV
Sysdata	Sysdata Jr.	Brasil	TRS-80 Mod.III
Timex	Timex 1000	USA	Sinclair ZX-81
Timex	Timex 1500	USA	Sinclair ZX-81
Timex	Timex 2000	USA	Sinclair Spectrum
Unitron	AP II	Brasil	Apple II +
Victor do Brasil	Elppa II Plus	Brasil	Apple II +
Victor do Brasil	Elppa Jr.	Brasil	Apple II +
Video Genie	Video Genie I	USA	TRS-80 Mod. I

UM LOGOTIPO PARA CADA MODELO DE COMPUTADOR

INPUT foi especialmente projetado para microcomputadores compatíveis com as sete principais linhas existentes no mercado.
Os blocos de textos e listagens de programas aplicados apenas a determinadas linhas de micros podem ser identificados por meio dos seguintes símbolos:

Sinclair ZX-81

TRS-80

TK-2000

MSX

Spectrum

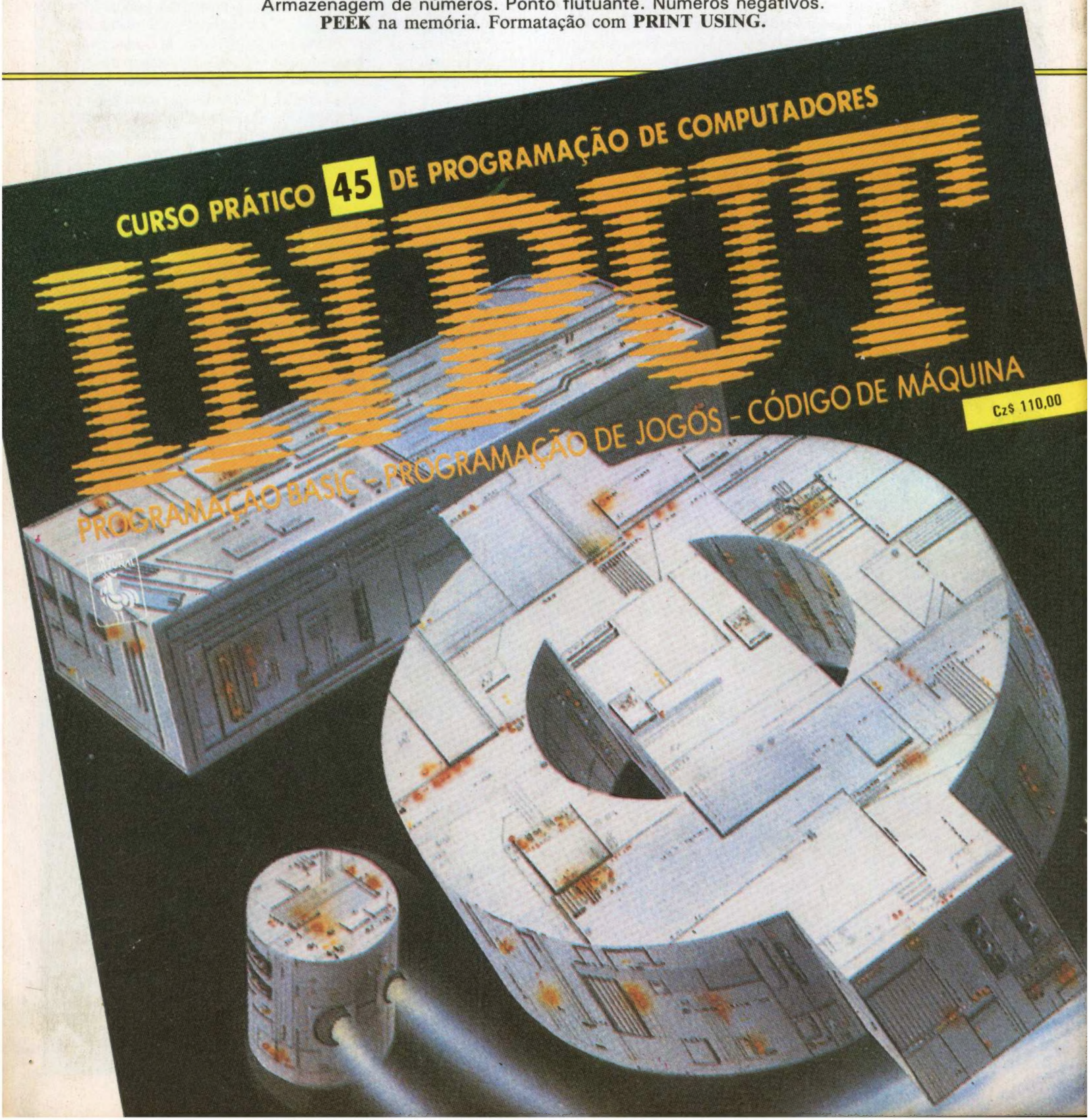
TRS-Color

Apple II

Quando o emblema for seguido de uma faixa, então tanto o texto como os programas que se seguem passam a ser específicos para a linha indicada.

NO PRÓXIMO NÚMERO

PROGRAMAÇÃO BASIC
Mensagens secretas: como produzir seus próprios códigos.
Códigos Morse, de Júlio César e de posição. Cifra Saint Cyr.

PROGRAMAÇÃO DE JOGOS
A rotina principal do caça-níqueis. Resultados premiados.
Como segurar, empurrar e apostar. Efeitos sonoros.

PROGRAMAÇÃO BASIC
Armazenagem de números. Ponto flutuante. Números negativos.
PEEK na memória. Formatação com **PRINT USING.**